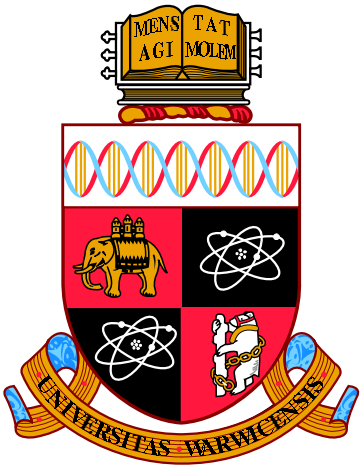


Temporal Incident Light Fields

by

Debmalya Sinha

Thesis

Submitted to the University of Warwick

for the degree of

Doctor of Philosophy

Warwick Manufacturing Group

June 2018

Contents

List of Figures

Acknowledgments

On my first day in England, Kurt came to pick me up; and he since has been, picking me up every time I'm lost wandering in a blind lane. Doctoral courses are not meant to be easy. They're testing, both with and without the pen. Alan made sure the ink never ran out, especially when I had to muster every last bit not to give up on that end-of-the-tunnel light. It would not have been possible to complete this work without my two advisors, Dr. Kurt Debattista and Prof. Alan Chalmers. I am truly fortunate and privileged to be their student.

"All the literati keep an imaginary friend" wrote W. H. Auden in his poem *"The fall of Rome"*. Fortunately, I had many tangible people around me to rely on whenever I reached out! I wouldn't have my spoon with me if it was not for Pinar and Rossella's train of advices on those many summer afternoons! My heartiest gratitude to Tom for all the brainstorming sessions throughout this time. Finally, Ratnajit, from those 2AM trips in freezing cold to the endless drafts and corrections of this thesis; I raise my glass to you brother.

I'd also like to thank Jon, Ali, Eva and everyone in the Sensomatics and the trueDR team for making it actually possible for me to write my thesis and continue working at the same time! Tons of sunny cheers to Tim, Amar, Josh, Martin, Mark, Emanuel and Demetris too for all the great memories and friendship throughout these years.

As many people said goodbye around me, all the buzzing afternoons around slowly started to turn into a quieter sombre. The evening would've crept in, if it wasn't for Pari and Kitim. Thank you grasshoppers, thank you butterflies, thank you, thank you pouring rains.

Finally, I can't help but remember my late grandmothers, Bela and Dheera, who would've been the happiest to see me now. I can see your smiling, slightly toothless faces every time I close my eyes. I wish you were here today.

Days end. Flocks come back to where they belong. Although for last two years, belonging became somewhat foggy for me; two people in a far off land were waiting with an undying belief, of which I perhaps am not worthy. All I am is because they taught me how. All my coming years will bask in their afterglow. This degree is a tiny step towards actually becoming the person that my makers, Ratna and Arun, hope I will someday become.

Declarations

The work in this thesis is original and no part of work referred to here has been submitted in support of an application for another degree or qualification of this or any other university or institute of learning.

Abstract

High-fidelity real-world lighting is a complex and rapidly expanding field of study in computer graphics. Rendering with real-world lighting plays a crucial part in motion pictures, computer games, Augmented Reality (AR) and Virtual Reality (VR) applications. There are, however, many constraints when capturing and representing real-world lights for rendering. In particular, dimensionality plays a significant role although existing industry-standard methods are inadequate to capture light throughout the three spatial, two angular and a temporal dimension simultaneously.

Image Based Lighting (IBL) techniques addresses temporality by capturing two angular and the temporal dimension simultaneously. The Incident Light Fields (ILF) technique, on the other hand, can capture complex spatially varying real-world light incident on a static scene covering five angular and spatial dimensions. However, any changes in the positions or the radiometric properties of direct light sources in the scene over time invalidates the captured ILF due to the subsequent changes in the indirect lighting. In a dynamically varying lighting condition, ILF needs to be recaptured with each change in the lighting which is infeasible in most real-world situations.

This thesis proposes a novel technique called "Dynamic Change Propagation" (DCP) that can simulate any changes made in the direct light and propagate the effects to the indirect lighting recorded in a captured ILF. Evaluations show average RMSE errors of 0.034 with absolute percentage errors of 6.8% for light source movement simulation and 0.013 (RMSE) for 3.4% for intensity change simulations.

In addition to the DCP technique, this thesis proposes a novel "Temporal Incident Light Field" (Temporal ILF) technique which records the changes in the light sources over time and utilizes the DCP technique to simulate those changes into the originally recorded static ILF thus, capturing six (spatial, angular and temporal) dimensions.

To the best of our knowledge, Temporal ILF is the first method which can record high-fidelity real-world light over all six spatial, angular and temporal dimensions simultaneously. The introduction of the DCP and Temporal ILF techniques in this thesis offers new ways of rendering with spatio-temporally variant high-fidelity real-world light.

Chapter 1

Introduction

"Begin every canvas with a wash of black, because all things in nature are dark except where exposed by the light."
– Leonardo da Vinci

Photorealism attempts to recreate the complexity and beauty of the real world in an image. Hiding the brushwork has been the pivotal effort of photorealism. Starting in the mid-twentieth century this has continued through various mediums, finally finding foothold in the virtual world through computer rendering. A key research area in Computer Graphics is to explore ways to produce images that are virtually indistinguishable from a photograph of a real scene or object. To achieve the desired accuracy requires three interconnected elements: (i) accurate geometry of the scene and precise material properties, (ii) high-fidelity light information, and (iii) physically-based rendering techniques which take the light and scene information as inputs to render photorealistic images. This thesis presents a novel real-world light representation technique called the "Temporal Incident Light Fields" (Temporal ILF) which can be used by physically-based renderers to illuminate synthetic objects in order to produce photorealistic renderings.

The Temporal ILF can capture and represent high-fidelity spatially varying real-world light which changes over time dynamically. None of the existing techniques can capture and represent a simultaneous spatial and temporal variation in lighting. Temporal ILF technique will enable various applications of Computer Generated Imagery (CGI) in motion pictures, Virtual Reality (VR) applications and object relighting [Masselus et al., 2003] all of which require dynamically changing real-world light with high accuracy. The following sections in this chapter describes the terminology, motivation behind this work, primary contributions and the organisation of this thesis.

1.1 Terminology

This section briefly describes the preliminary concepts and terminologies used throughout this thesis.

1.1.1 Photorealistic Image Synthesis

Physically-based rendering (PBR) is a technique in which the light transport in a virtual scene is calculated to resemble the interaction of light in a real-world scene as accurately as possible [Pharr et al., 2016]. High-fidelity inputs and physically-based rendering are the key components of a photorealistic render. In general, there are two types of inputs i.e. (i) a mathematical model of 3D geometry with the material properties of the object/scene to render and (ii) a high-fidelity lighting information of the environment. Advanced 3D modeling softwares, such as Maya, 3DS Max, Blender, etc. can produce very accurate 3D models of objects. Moreover, there are existing methods [Levoy et al., 2000; Rusinkiewicz et al., 2002; Izadi et al., 2011] that can "scan" real objects to obtain its 3D geometry and material properties such as colour and light reflectivity with reasonable accuracy. On the other hand, there are several techniques such as Image Based Lighting (IBL) [Debevec, 2002], Lumigraph [Gortler et al., 1996], and Incident Light Fields (ILF) [Unger et al., 2003] that can capture and represent the real-world light in an environment with varying degrees of dimensionality and capabilities. The work described in this thesis will focus on the real-world light information inputs.

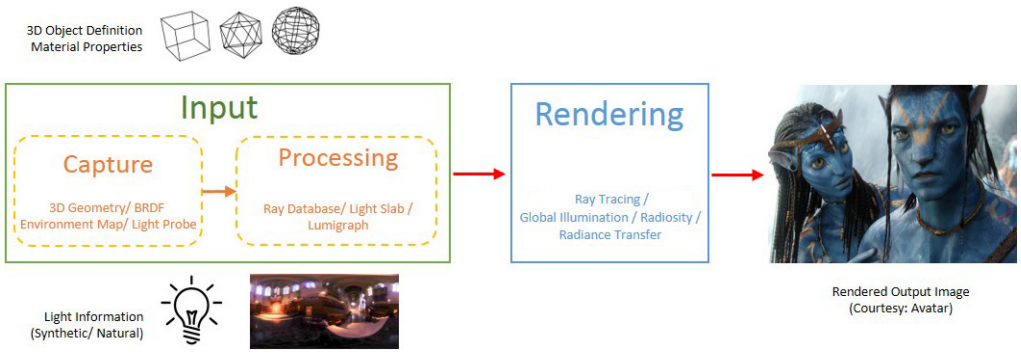


Figure 1.1: A general flow chart for photorealistic image synthesis

Figure 1.1 shows the pipeline for photorealistic rendering. Both types of inputs, the scene geometry and lighting information, go through the steps shown in the pipeline; i.e. capture, processing and rendering. Firstly it is necessary to capture the inputs with required precision. Secondly the captured inputs are represented in the appropriate format for the renderer. Lastly the represented inputs are used by PBR techniques to produce photorealistic renderings.

As the accurate reproduction of real-world lighting, which varies spatially and changes over time, is the primary focus of this thesis; the discussion of inputs to the renderer will be confined to lighting representation from here after.

1.1.2 Applications of Photorealistic Rendering

Accurate reproduction of real-world lighting scenarios have a significant impact on various virtual reality applications such as the automotive, healthcare or engineering simulations or various Computer Generated Imagery (CGI) applications including animations and feature films. Below are the few applications of high-fidelity rendering of real life scenes.

1.1.2.1 Motion Picture

The motion picture is one of the largest industries where there is a constant demand for photorealistic renderings. Currently Computer Graphics is so closely related to the film and entertainment industry that there are separate Academy Awards (Oscar) for novel technical contributions in motion pictures. In addition to the animated films by Disney and Pixar labs, films such as "The Matrix" or "Avatar" implemented many novel CGI techniques which set the standards for almost all the movies in Action or Science fiction genre that followed.

1.1.2.2 Computer Games

The other most significant and financially lucrative industry that extensively uses photorealistic rendering is the computer games industry. The revenue in computer games (software only) industry worldwide in 2018 was 89 billion USD which is steadily increasing over all the segments such as console, mobile and VR games [gam]. Modern games are not only about the story, but the whole immersive experience of the gameplay with real time high definition photorealistic video. The advances in GPU technology has made it possible for games to provide highly sophisticated real time 3D renderings of scenes often reproduced with Virtual Reality display hardwares such as Oculus Rift and HTC ViVE. Photorealism plays a huge part in the believability of the rendered images in computer games.

1.1.2.3 Flight and Car Simulations

The flight and car simulators used for testing and training purposes benefit heavily from high-fidelity light and atmospheric conditions. The accurate reproduction of light outside the car is often crucial for inexpensive driver-experience testing without actually driving in a test track which could be much more time consuming, safety critical and expensive. Apart from exterior lighting the simulation of interior lighting

in driving simulators is of utmost importance [Debattista et al., 2017]. Armed combat, sports and other niche applications of simulators also benefit from photorealistic rendering.

1.1.2.4 Architecture

Photorealistic renderings in architecture has similar benefits to simulators. How the insides of a room will look like on a sunny day or in dull conditions, how many and what type of artificial lights will be needed to get the desired look inside a room etc. can only be determined with a photorealistic rendering using high-fidelity light information. A second application within the architecture genre resides with rendering cultural heritage architecture installations displayed in a head-mounted display which deals with photorealistic rendering of the age old architecture and sometimes even rendering parts of it that has been destroyed over time.

Although only a few applications are mentioned here, there are many more diverse applications of photorealistic rendering. New fields such as various AR and VR related fields are emerging rapidly and the demand of high-fidelity lighting is growing at a steady pace.

1.1.3 Real-World Lighting

A light ray is represented by its origin position and its direction. In a real-world dynamic scene, light changes over time, thus, real-world lighting can be represented with a 7-dimensional function $P = P(x, y, z, \theta, \phi, t, \lambda)$ called the plenoptic function [Adelson and Bergen, 1991]. Details about this can be found in Chapter 2. The (x, y, z) are the three dimensions of space, (θ, ϕ) is for direction, t is for time and the λ is for the wavelength which is usually ignored. Various real-world light capture and representation techniques attempts to cover subsets of these dimensions.

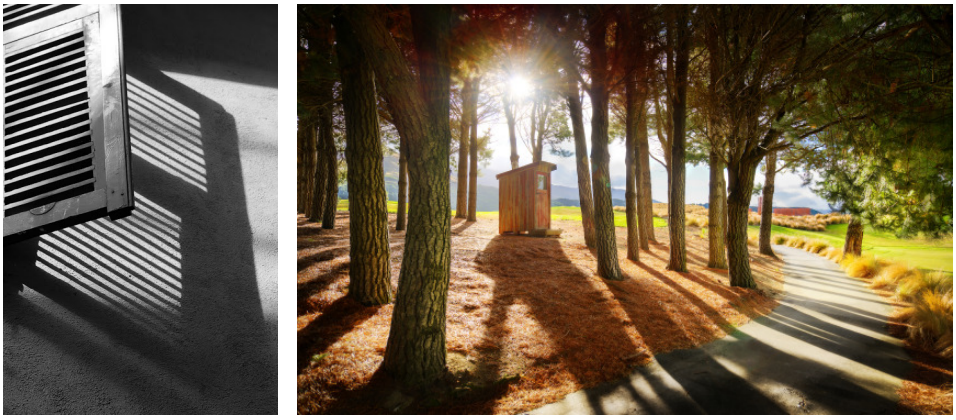
Real-world light is captured using High Dynamic Range (HDR) [Banterle et al., 2017] photographs of the scene which captures detailed information in the shadow and the highlight regions effectively. Chapter 2 describes more about HDR images and capturing techniques in detail. An HDR image of the surrounding environment is called the HDR *environment map*. When an environment map covers the entire upper hemisphere of any given plane with $(360^\circ \times 180^\circ)$ angular resolution, it is called a *light probe*.

The light probes are used either directly as light sources [Debevec, 2002], or processed into a secondary data structure such as a ray-database [Unger et al., 2008] which then is used as the light source for a physically-based renderer to render synthetic objects with real-world light.

1.1.3.1 Image Based Lighting (IBL)

IBL is a widely used method for photorealistic rendering using real-world light [Debevec, 2002]. It uses the environment maps of real scenes as the light source for direct and ambient indirect light. IBL is particularly useful for rendering synthetic objects in a real world scene. Chapter 2 describes IBL in more detail. Despite its simplicity and effectiveness, the major drawback of IBL is it only covers the two angular dimensions (θ, ϕ) in P and not the spatial dimensions, thus assuming the incident light on the rendering space is uniform, which in reality is often untrue.

1.1.3.2 Spatially and Temporally Varying Lights



(a) (b)

Figure 1.2: Spatially varying lighting in real-world.

Lights in the real-world are non-uniform and dynamic in most cases. The intensity and colours in a scene often vary spatially, such as the scene shown in Figure 1.2 displaying *spatially varying light*. Moreover, the lights can change dynamically when the light source changes position or the intensity and colour over time. For instance, sun changes its position throughout the day and so does the sunlight accordingly. On the other hand, artificial light sources can be moved, dimmed or completely switched off and on dynamically over a time period. This is the *temporal variation* in light and is particularly useful in rendering videos where the light sources keep changing.

1.1.4 Brief Introduction to the ILF

State-of-the-art techniques for capturing a real environment such as IBL assumes the incident light from the environment into a scene is uniformly distributed. To capture such uniform lighting, a single environment map is sufficient as shown in the

left table in Figure 1.3 where it is uniformly lit and a single light probe has been taken in the place of the red dot. The blue arrows represent light coming from the environment all around the upper hemisphere of the table.

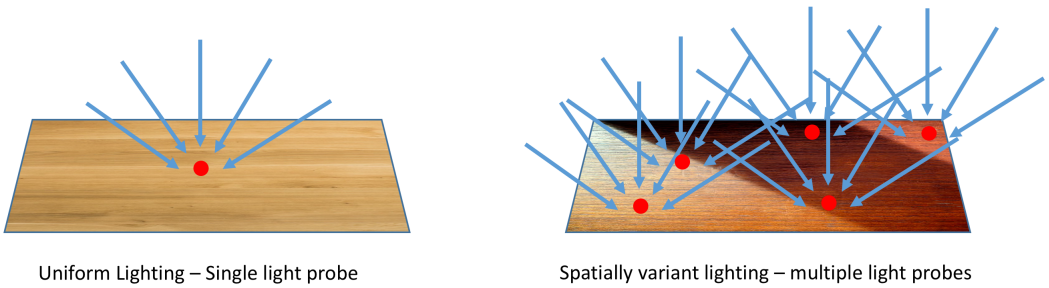


Figure 1.3: Single environment map for uniform lighting in IBL versus multiple environment maps while capturing non-uniform, spatially variant lighting.

Real-world lights however often vary spatially. For instance, the sun shining through a window blind would result in strips of light and shade as shown in the right side table in Figure 1.3. ILF Unger [2009] provides means to produce high-fidelity renderings for spatially varying light by capturing multiple light probes throughout the space at every possible point. Each pixel on the light probes represent a particular spatial point and its corresponding angular direction, thus capturing 5 dimensions (x, y, z, θ, ϕ) out of possible 6 in P. The light probes are stored in a special data-structure to quickly access the lighting information given a tuple of the location of a point in the scene and a direction from the point towards the environment. Chapter 4 describes ILF in much greater detail.

1.1.5 Limitations of Current Techniques

Despite being able to represent high-fidelity spatially varying light, there are two major limitations of ILF.

Firstly, ILF only renders static scenes and is incapable of recording and rendering scenes where the light information changes over time. As real-world light varies spatially and temporally simultaneously, there are many real-world video rendering applications that cannot use ILF because of this limitation and use the Temporal IBL [Havran et al., 2005] instead, sacrificing the spatial variations.

Secondly, there are options that can change the intensity and colour of a light source in ILF by directly manipulating the part of the ray database that contains lights from that light source. However, the changes made to the light sources do not propagate to the entire ILF ray-database, e.g. indirect lighting from the captured scene geometry. For example, if the only light source in a captured scene is turned off (change intensity to zero) in ILF database post processing, the indirect light that

has been stored in the ILF database should also be affected by it. Traditional ILF cannot propagate this change.

1.2 Research Objective

The objective of this Ph.D. is to develop a high-fidelity light representation technique for spatio-temporal variations in real-world lighting.

The existing Incident Light Field (ILF) technology provides a high-fidelity rendering solution for spatially varying lighting. This Ph.D. describes a novel technique called the "Temporal Incident Light Fields" (Temporal ILF) which augments the ILF technology further to accommodate the ability to record, store, and process temporal variations in spatially varying real-world light and render synthetic scenes from it.

1.3 Primary Contributions

The primary contributions of this thesis are:

1. **Implementation of ILF:** Although the basic principle is same, ILF has a few different implementation techniques in terms of how its ray-database is constructed from the collection of the captured light probes. The "free-form ILF" [Unger et al., 2008] is the most versatile technique among all. The research in this thesis has further modified the free-form ILF to better suit the novel Temporal ILF technique. The implemented version in this work changes a few representational aspects from the original method which does not affect the precision of the original ILF.

2. **Dynamic Change Propagation:** As mentioned in Section 1.1.5, one of the major limitations of ILF is its inability to propagate the effects of any changes made in the light sources to the indirect lighting. An ILF capture of a real-world scene contains light information from the direct lighting and the indirect lighting reflected from the scene geometry both of which can be stored separately in the ILF ray-database. This work describes a novel technique called the "Dynamic Change Propagation" (DCP) which can change the position or radiometric properties of the direct light sources synthetically in an ILF database and subsequently propagate the effects of these changes to the indirect light information stored in the ILF database, thus making the synthetic changes realistic and physically accurate.

3. **Development of Temporal ILF:** Temporal ILF is the novel technique for capturing and render from spatio-temporally varying light. It however cannot

be captured by naively capturing multiple snapshots of static ILFs over a period of time because a single snapshot of the ILF capture is not instantaneous and can take a significant amount of time depending on the method. Even the current fastest method of capture by moving a HDR video light probe is not feasible for capturing over time. Temporal ILF implements a unique capturing scheme where one static ILF is captured and subsequently the position and radiometric changes of the changing light sources are recorded over time. The captured static ILF is then synthetically changed with the DCP technique for each recorded change of the light sources to obtain multiple snapshots of the ILF over time, thus acquiring the Temporal ILF.

4. **Evaluations:** The works described here has been evaluated with quantitative comparison of a ground truth images with the rendered image. Several metrics has been used to determine the accuracy of the DCP technique as well as the accuracy versus computation time for different setups of it.

1.4 Thesis Outline

This chapter provides an introduction to high-fidelity photorealistic image synthesis and its applications in general. It very briefly introduces the relevant existing approaches to achieve a high-fidelity representation of real-world light and the current limitations. Lastly it provides the research objectives for this thesis, the motivations and their possible applications. The rest of the chapters in this thesis are organised as follows:

Chapter 2 serves as the background for various concepts used in high-fidelity rendering. It describes concepts from physically-based quantities like Radiometry and various reflection functions (BRDF), the necessary concepts of light transport theory such as the rendering function, monte-carlo methods, Global Illumination (GI), ray tracing, path tracing and related concepts. Lastly it describes the various techniques for real-world light representation such as IBL, Lumigraphs, Light fields and its few variations. These concepts are used throughout the development of the work done in this thesis.

Chapter 3 gives a high level concise overview of the workflow of this thesis. There are three distinct parts of work that has been described in this thesis which eventually aids to develop the Temporal ILF. This chapter describes the brief methodologies and working principles of (i) the ILF Implementation, (ii) the Dynamic Change Propagation, and finally (iii) the Temporal ILF in a nutshell to elucidate the motivation for each of them and the interconnection in between.

Chapter 4 provides a detailed description of the ILF technique, the ILF pipeline, various methods of ILF capture, the processing of the captured environment maps into ray-databases and lastly the rendering technique with ILF. In addition to the original working principle of the ILF, this chapter describes the ILF implementation that has been specifically developed for this work which changes a few aspects of the original method to suit the purpose of eventually developing the Temporal ILF.

Chapter 5 describes the Dynamic Change Propagation technique in detail and how it can be used to change the ILF ray-Database realistically. This chapter also provides the evaluation of the technique by calculating RMSE and Absolute percentage errors and tests the computation time versus its accuracy using two virtual environments.

Chapter 6 gives a detailed description of the Temporal ILF, its pipeline, captures, processing and the rendering. It also introduces two possible methods for capturing the Temporal variation one of which was used to implement a real-world implementation of the Temporal ILF.

Chapter 7 concludes this thesis with a general discussion, limitations and possible future works that can be done with appropriate resources.

Chapter 2

Rendering with High-Fidelity Lighting

"Not the old, not the new, but the necessary"

– Tristan Tzara

Rendering is the technique of generating a two dimensional image from a three dimensional description. This chapter describes ways to achieve "photorealistic" renderings with high fidelity lighting which are indistinguishable from a photograph. The central part of this process is *Physically-Based Rendering Techniques (PBRT)* which simulate reality. Photorealistic rendering requires (i) mathematical representation of the scene geometry, (ii) Physically based representation of light and materials, and (iii) Physically based light transport techniques. The work described in this thesis concentrates on the light representation technique. This chapter will provide a background on the various concepts of *physical light transport* and the relevant *physically-based rendering techniques* i.e. how the light interacts with the different areas of the scene realistically and calculating the amount of light each point in the scene receives. In addition to that, this chapter also describes the relevant mathematical representations of *high fidelity light information* which are ideally acquired from real-world environments. As shown in Figure 2.1, the sections in this chapter can be divided into three interconnected sets which in tandem produces a photorealistic rendering of a synthetic scene from real-world light.

The first group is *Physically-based quantities* where Section 2.1 gives an introduction to Radiometry and its various quantities while Section 2.2 describes the interaction of light with various media, i.e. reflection and refraction distribution functions [Dutre et al., 2006].

The second set of sections describe the *Physically-Based light transport theory* which is the core part of Physically-Based Rendering Techniques (PBRT)[Pharr et al., 2016] where all these physically-based quantities and light transport theo-

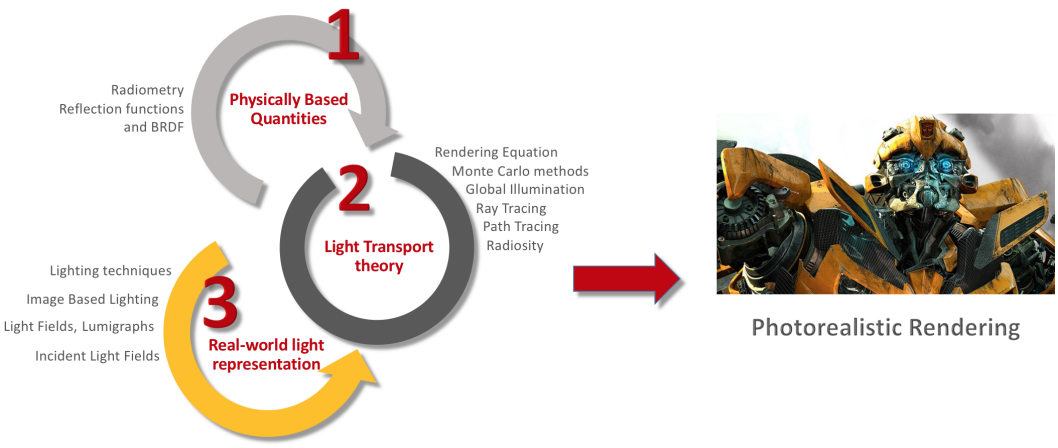


Figure 2.1: Interrelations of the sections in this chapter. Image from screen capture of the motion picture, "The Transformers"

ries are implemented to render an image. Starting with Section 2.3 the "rendering equation" [Kajiya, 1986], it briefly introduces Monte-Carlo techniques in Section 2.4 which is useful to find the solution of the rendering equation. Finally, Section 2.5 describes Ray Tracing, Path Tracing and Radiosity as part of various Global Illumination algorithms [Dutre et al., 2006].

The third and final set of sections describe the *Real-word light representation* which encapsulates various techniques for capturing and rendering from high fidelity real-world light which is crucial to photorealistic rendering, as in many cases virtual objects are rendered in a real scene [Debevec, 2008]. Section 2.6 introduces to background of lighting techniques including Image Based Lighting (IBL)[Debevec, 2002]. Section 2.8, 2.9 and 2.9.4 subsequently describes more complex lighting techniques such as Lumigraph [Gortler et al., 1996], Light Fields [Levoy and Hanrahan, 1996] and Dynamically re-parameterised Light Fields [Isaksen et al., 2000] respectively.

This chapter concludes with a discussion in section 2.10 about all the necessary techniques discussed and provides a segway to the next chapter which discusses Incident Light Fields which is the bedrock of the work done in this thesis.

2.1 Radiometric Quantities

The rendering algorithms compute the steady state distribution of light energy in a scene; i.e. given a 3 dimensional (3D) scene and a light source, they compute the amount of outgoing light from a particular point in a particular direction. Radiometric quantities described in this section are the physical quantities that represent the amount of light. The following descriptions are useful to eventually derive the rendering equation.

2.1.1 Evolution of light models

Since the Newtonian times, there has been a few different models of light, most of which are used in modern times for different types of rendering. The most rudimentary one is the *Geometric model* which states that light travels instantaneously in a straight line, and is not affected by other forces such as gravity [Hamilton, 1828]. James Clerk Maxwell's wave equations changed our notion of energy and thus a new *Wave model* of light was formed [Maxwell, 1865]. Later, during the early 1900s, Max Plank's work on how energy travels in quanta explained the wave-particle duality of energy [Planck, 1901]. Five years after that Albert Einstein showed that these proposed quantum of light, named photons, have energy $E = \hbar\nu$ where ν is the frequency of the light and $\hbar$ is Planck's constant [Einstein, 2005; Arons and Peppard, 1965]; thus marking a new, *Quantum model* of optics. Later, as the understanding of the building blocks of energy and matter became more and more clear (obscured), there are revisions into quantum optics in the forms of Quantum Field Theory and Quantum Electro-Dynamics [Feynman and Zee, 2006].

All of these models are relevant to computer graphics for their uses in various different type of rendering algorithms [Pharr et al., 2016]. For example, rendering caustics are much efficient with the quantum model of light. However, the vast majority of the work in computer graphics is based on the simplest, geometric model of light and the associated measures or radiometric quantities; Flux, Irradiance and Radiance. The following sections will thus assume the geometric model hereafter.

2.1.2 Flux

The most fundamental radiometric quantity is flux (Φ). This is the total energy flowing (incident or emitted) through a surface per second. The unit for Flux is Watt(W) or (Joule/second).

2.1.3 Irradiance

Irradiance (E) is the flux per unit square area, i.e. the total radiant energy incident per second per square unit surface. Irradiance is calculated as $E = \frac{d\Phi}{dA^{\perp}}$. The unit of irradiance is Watt/meter^2

2.1.4 Radiance

Radiance, denoted by L, is the flux per unit projected area per unit solid angle. Radiance calculates the light energy incident per second through an unit surface coming from a point. Figure 2.2 explains the components of radiance. In the diagram, P_1 is the light source from which light energy is incident on the unit surface area (A) measured in meters around point P_2. The red arrow is the surface normal of A on

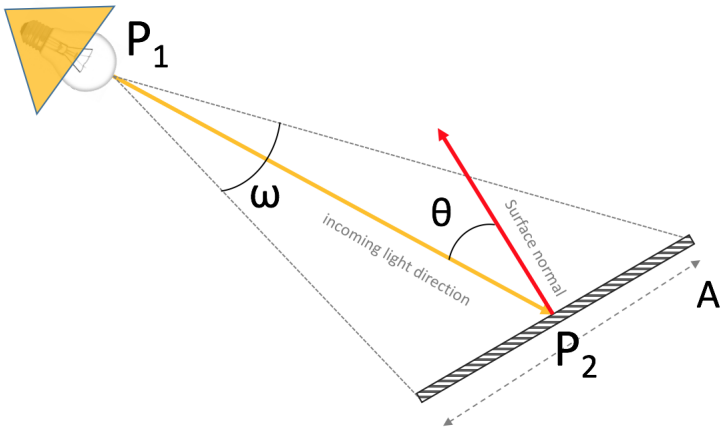


Figure 2.2: The relations of light flow rate with the area A, solid angle ω and angle of incidence θ. The incident light follows yellow arrow on the surface while the red arrow is the direction of the surface normal of the surface A.

point P_2. The incidence angle of light from P_1 to P_2 is the angle θ. The solid angle for the total light energy coming to the unit area A around P_2 is ω which is measured in steradian. the The unit of Radiance is $Watt/steradian * m^2$.

$$L = \frac{dE}{d\omega} = \frac{d^2\Phi}{d\omega dA^{\perp}} = \frac{d^2\Phi}{d\omega dAcos\theta} \tag{2.1}$$

The next section will briefly describe the interaction of light with various types of material surfaces.

2.2 Interaction with Surfaces

Assuming the geometric model of optics, light interacts in three possible ways after contact with any matter, i.e. reflection, refraction and absorption. Different materials reflect, refract and absorbs different amount of light which can be as simple as pure reflections in a mirrored surface to much more complex reflections which adds components of diffraction and scattering. Ignoring the complex interactions for brevity, reflection happens in broadly three ways for most materials; *diffused*, *specular* and a mixture of these two, termed as *glossy* in Computer Graphics terminology. The following sections will describe the mathematical representation of the light interaction.

2.2.1 Bi-directional Reflection Distribution Function (BRDF)

The different types of reflections can be estimated by a function which, given a specific radiance amount L_i in a particular direction ω_i, calculates the radiance L_o reflected back from the surface in the given direction ω_o. This function is called

the Bi-directional Scattering Surface Reflection Function (BSSRDF). However, assuming that the light incident on a point $\mathbf{x}$ on a surface reflects from the same point $\mathbf{x}$ simplifies the relation by ignoring the more complex, sub-surface scattering phenomenon. The new simplified function is called Bi-directional Reflectance Distribution Function (BRDF) [Bartell et al., 1981]. The BRDF at a point $\mathbf{x}$ is given by the function $f_r(x, \Psi \rightarrow \Theta)$ where Ψ and Θ are the incident and reflected ray direction respectively. In the following equation, the ω is the solid angle between light source and the area similar to the diagram in Figure 2.2.

$$f_r(x, \Psi \rightarrow \Theta) = \frac{dL(x \rightarrow \Theta)}{dE(x \leftarrow \Psi)} = \frac{dL(x \rightarrow \Theta)}{dL(x \leftarrow \Psi) cos\theta d\omega} \tag{2.2}$$

BRDF is the ratio of the outgoing radiance $L(x \rightarrow \Theta)$ and the incoming irradiance $E(x \leftarrow \Psi)$ at the point $\mathbf{x}$. Equation 2.2 further decomposes the irradiance in terms of incoming radiance at an angle θ with the surface normal at point $\mathbf{x}$. The incoming radiance is calculated over the entire hemisphere around the point $\mathbf{x}$. In practice, the hemisphere is discretised into blocks of solid angles $d\omega$ [Dutre et al., 2006]. The total radiance is computed by integrating the discreet radiances over the full hemisphere.

2.2.2 BRDF Properties

There are four main properties of a BRDF as described below.

2.2.2.1 Anisotropy

Anisotropy is one of the main properties of BRDF. It is a seven dimensional function of 3D position x, outgoing ray angle Θ, and incoming ray angle Φ. $f_r(x, \Theta, \Psi) = f_r((x, y), (\theta_i, \phi_i), (\theta_o, \phi_o))$. A more detailed representation uses the wavelength (λ) of the incoming light but it is ignored here for brevity. Changing any of these dimensions naturally changes the BRDF too. For example, if the surface containing point x is rotated around the surface normal on x, as the incident angle changes, the BRDF will also change. However there are a few materials that are *isotropic* and this kind of rotation has no effect on their reflective properties.

2.2.2.2 Helmholtz Reciprocity

The Helmholtz reciprocity dictates that reversing the direction of light does not change the amount of reflected light [Von Helmholtz, 1867].

$$f_r(x, \Psi \rightarrow \Theta) = f_r(x, \Theta \rightarrow \Psi) \implies f_r(x, \Theta \leftrightarrow \Psi) \tag{2.3}$$

Therefore, both the direction in the BRDF function can be used interchangeably.

2.2.2.3 Linearity

Linearity is one of the primary properties of BRDF which states that the value of the BRDF for a particular incident direction is independent of any number of irradiance along other incident directions on the same point. BRDF acts as a linear function with respect to all possible simultaneous incident directions and thus is integrable over the hemisphere surrounding the point **x** [Bartell et al., 1981].

$$dL(x \rightarrow \Theta) = f_r(x, \Psi \rightarrow \Theta) dE(x \leftarrow \Psi)$$

Integrating over hemisphere (Ω) around **x**,

$$L(x \rightarrow \Theta) = \int_{\Omega} f_r(x, \Psi \rightarrow \Theta) dE(x \leftarrow \Psi) \tag{2.4}$$

This is especially useful while deriving the "Rendering Equation" as described in section 2.3.

2.2.2.4 Energy conservation

The final property is conservation of energy which is central to the formulation of Physically-Based BRDF. It ensures that the incoming energy is always exactly same to the sum of total outgoing energy and the absorbed energy. Expanding equation 2.2, the ratio of incoming and outgoing radiance is always ≤ 1 [Dutre et al., 2006][Bartell et al., 1981].

$$\frac{dL(x \rightarrow \Theta)}{dL(x \leftarrow \Psi)} = f_r(x, \Psi \rightarrow \Theta) cos\theta d\omega \leq 1 \tag{2.5}$$

The next three sections briefly describe the most commonly occurred BRDF types, diffused, specular and the glossy, respectively.

2.2.3 Diffused and Specular BRDF

Diffused BRDF: Diffused surfaces are made of materials that reflect light uniformly over the entire hemisphere around any given point **x** on the surface. Regardless of the incident angle of light, a point in a purely diffused surface looks exactly the same from all directions around it. An Ideal diffused BRDF is given by $f_r(x, \Psi \leftrightarrow \Theta) = \rho_d/\pi$ where ρ_d denotes the reflectance, i.e. the fraction of incident energy reflected on point **x**.

Specular BRDF: Incident light at a given point on Specular surfaces reflects and refracts light only in one particular direction. Following the geometric model of optics [Hamilton, 1828], given the incident angle (Ψ_i) and the surface normal ($\hat{N}_x$) on the point of incident, the reflection direction (R) can be calculated as:

$$R = 2(\hat{N}_x \cdot \Psi_i)N - \Psi_i \tag{2.6}$$

Light can pass through mediums such as air, glass and water. In the process, the direction Θ_r in the refractive medium of the light changes from the incident direction (Ψ_i) (in the incident medium) on the point of contact between the two mediums. Snell's law [Feynman et al., 2011] gives us the relation as:

$$\eta_r sin\Theta_r = \eta_i sin\Psi_i \tag{2.7}$$

Here, the constants η_r and η_i are called the refractive index and they belong to the refractive medium and the incident medium respectively. Usually η is same for a given medium but in some cases it can vary throughout the medium causing complex phenomenons such as sub-surface scattering [Hanrahan and Krueger, 1993].

2.2.4 Glossy BRDF

Both diffused and specular BRDFs are mostly theoretical because real-world materials show a degree of ideal specular or ideal diffused BRDFs. Often materials have a combination of diffused and specular BRDFs. Such kind of surfaces are called glossy surfaces. Due to the apparent complexity of modeling BRDF for real-world objects, there are a number of shading models proposed that can replicate different types of glossy surfaces. This section provides a discussion on the 5 most relevant BRDF models out of them.

2.2.4.1 Lambertian model

The Lambertian model [Lambert, 1760] is the simplest and oldest of all models. Constructed around 1760 by Swiss mathematician Johann H. Lambert, it models the diffused BRDF where the value of BRDF is constant ($f_r(x, \Psi \leftrightarrow \Theta) = k_d = \rho_d/\pi$) over any direction of the hemisphere over $\mathbf{x}$ [Bartell et al., 1981; Dutre et al., 2006].

2.2.4.2 Phong model

Phong is the most popular shading model to date partly because it is slightly less complicated than some of the other models and it is hardware accelerated. The

BRDF is given by:

$$f_r(x, \Psi \leftrightarrow \Theta) = ks\frac{R \cdot \Theta^n}{\hat{N} \cdot \Psi} + k_d \tag{2.8}$$

Here, reflected ray direction R can be found with Equation 2.6 and k_d is found from the lambertian constant BRDF ρ_d/π. All the other symbols bear usual references. However, Phong model does not support energy conservation or Helmholtz reciprocity as a tradeoff for its simplicity and the less computational requirements [Phong, 1975].

2.2.4.3 Modified Blinn-Phong model

Modified Blinn-Phong model [Blinn, 1977] solves the issues with Phong model and uses the half vector (**H**) between incident direction Ψ and eye direction Θ.

$$f_r(x, \Psi \leftrightarrow \Theta) = ks(\hat{N} \cdot \mathbf{H})^n + k_d \tag{2.9}$$

All the symbols used have the same references as the previously described models. Modified Blinn-Phong is a fairly popular model in Computer Graphics.

2.2.4.4 Cook-Torrance model

The Modified Blinn-Phong model was further augmented by Cook and Torrance to introduce a micro-faceted model [Cook and Torrance, 1982]. It assumes that the surface is made of randomly placed and oriented micro-sized smooth planer surfaces. Given a distribution of the facets, this model can model real-world objects fairly closely.

$$f_r(x, \Psi \leftrightarrow \Theta) = \frac{F(\beta)}{\pi}\frac{D(\theta_h)G}{(\hat{N} \cdot \Psi)(\hat{N} \cdot \Theta)} + k_d \tag{2.10}$$

Here "F" is the unpolarised Fresnel reflectance calculated over the angle β [Kenyon, 2008], D is the micro-facet distribution and the G denotes a geometry term for shadowing between those micro-facets.

2.2.4.5 Ward model

Gregory Ward took a new approach based on empirical data [Ward, 1992]. Ward and other comparable models such as LaFortune [Marschner et al., 1999] are the closest to the BRDF of real-world objects because of the intuitive parameterisation for the amount of diffused reflectance (ρ_d), specular reflectance (ρ_s) and the measure of roughness of the surface α. All three parameters can be changed at will and this

will still hold the energy conservation law.

$$f_r(x, \Psi \leftrightarrow \Theta) = \frac{\rho_d}{\pi} + \rho_s \frac{e^{\frac{-tan^2\theta_h}{\alpha^2}}}{4\pi\alpha^2\sqrt{(\hat{N}\cdot\Psi)(\hat{N}\cdot\Theta)}} \tag{2.11}$$

A number of other empirical models with many different approaches exist, such as models that fit empirical data obtained from real-world materials to known distributions and they all have their own set of advantages. However, assuming the general notion of the radiometric quantities and how light reacts with the materials can be used to model the equilibrium of light energy in a scene. Next Section 2.3 describes the "rendering equation" as a mathematical formulation of the energy equilibrium in a scene.

2.3 The Rendering Equation

The previous sections highlighted the physics of light transport; i.e. the way light interacts with the various types of materials. A Physically-Based system computes the light transport inside a scene by solving the primary light transport equation - the "rendering equation".

The rendering equation (light transport equation) describes the energy equilibrium in a scene. Using an approach similar to the radiative heat transfer equations, James Kajiya derived the equation from the simple notion that the total intensity on a surface point $\mathbf{x}$ is the summation of the total emitted intensity from $\mathbf{x}$ and the incoming intensity at $\mathbf{x}$ from every direction over the hemisphere around $\mathbf{x}$ [Kajiya, 1986]. Thus, total outgoing Flux $\Phi_o(x)$ at point $\mathbf{x}$ will be the sum of total emitted flux $\Phi_e(x)$ and total incoming flux $\Phi_i(x)$ multiplied by the reflectance factor on point $\mathbf{x}$, $C(x)$ $(0 \leq C(x) \leq 1)$. The following section will describe the formulation of the Rendering Equation.

2.3.1 Formulation

According to the energy conservation law:

$$\Phi_o(x) = \Phi_e(x) + C(x)\Phi_i(x)$$

Converting flux to irradiance by taking derivative w.r.t area A around point $\mathbf{x}$.

$$\frac{d\Phi_o(x)}{dA} \implies E_o(x) = E_e(x) + C(x)E_i(x) \tag{2.12}$$

Rewriting as an integral of all directions over hemisphere (Ω):

$$E_o(x) = \int_\Omega L_e(x \to \omega)d\omega + \int_\Omega C(x \leftarrow \omega)L_i(x \leftarrow \omega)d\omega \tag{2.13}$$

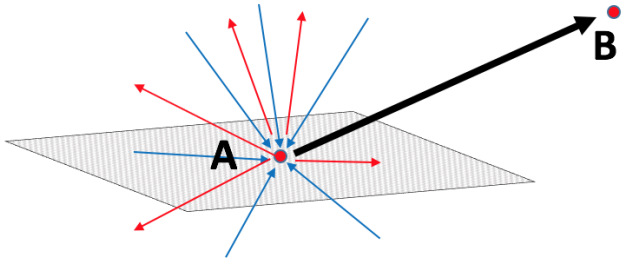


Figure 2.3: Transported light intensity from point A to B is the sum of all the incoming light at A and the emitted light from A towards B.

Equation 2.13 can be re-written as the equation 2.12 as an integral of radiance over all possible incoming angles over the hemisphere. $L(x \leftarrow \omega)$ is the radiance incoming towards x in the direction ω. Similarly, $L(x \to \omega)$ is the radiance emitting from x along the direction ω. This introduces the concept of light flow along different directions. Accordingly, the reflectance factor $C(x)$ from equation 2.12 has changed into $C(x \leftarrow \omega)$.

The final goal is to calculate outgoing light flow (outgoing radiance) from the point **x** to a certain point **C** in the direction ψ. Figure 2.4 shows a visual schematic of the same. Taking derivative of $E_o(x)$ with respect to outgoing angle ψ calculates the outgoing radiance towards point **C**.

$$\frac{dE_o(x)}{d\psi} = \int_\Omega \frac{L_e(x \to \omega)}{d\psi}d\omega + \int_\Omega \frac{C(x \leftarrow \omega)}{d\psi}L_i(x \leftarrow \omega)d\omega \tag{2.14}$$

From equation 2.1, $\frac{dE}{d\psi}$ is the outgoing radiance $L_o(x \to \psi)$. Moreover, intuitively, $\int_\Omega L_e(x \to \omega)\frac{1}{d\psi}d\omega$ is the emitted radiance from **x** towards direction ψ. Thus,

$$L_o(x \to \psi) = L_e(x \to \psi) + \int_\Omega \frac{C(x \leftarrow \omega)}{d\psi}L_i(x \leftarrow \omega)d\omega \tag{2.15}$$

Recalling equation 2.2, the ratio of the outgoing and the incoming radiance is, by definition, called the BRDF $f_r(x, \psi \leftarrow \omega)$. Compensating for the angle θ of the surface normal N_C of point **C** and the direction $(-\omega_o)$ as shown in figure 2.4,

$$L_o(x \to \psi) = L_e(x \to \psi) + \int_\Omega f_r(x, \psi \leftarrow \omega)L_i(x \leftarrow \omega)(N_A \cdot -\omega_o)d\omega \tag{2.16}$$

The above mentioned rendering equation is widely used in almost every type of

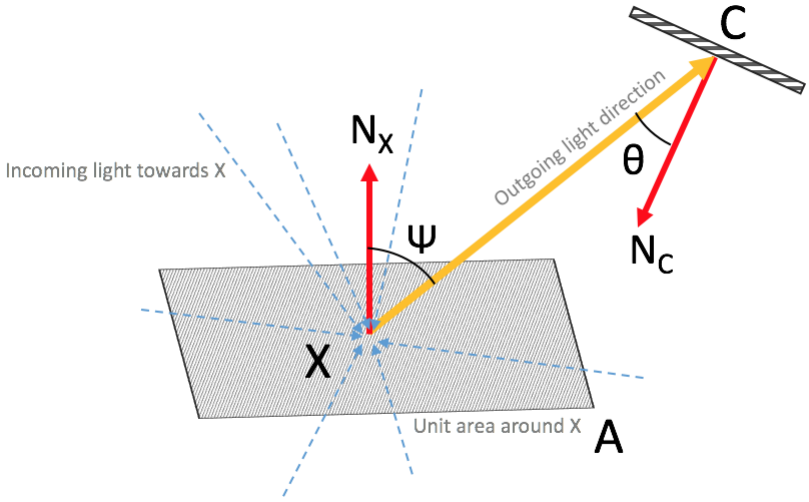


Figure 2.4: The light ray in yellow is coming from the point X towards the point C (usually the camera position) with an angle ψ with respect to the red Surface normal of the unit area A around point X. The blue arrows indicate incoming light rays from all around the environment towards point X.

Physically-Based rendering techniques because it satisfies the energy conservation laws and accurately approximates the light energy flow from one surface point to another. Next section will describe the various relevant Physically-Based rendering methods used to render photorealistic images.

2.3.2 Solving the Rendering Equation

The rendering equation is essentially an integral of the form

$$\phi(x, y) = \phi(x, y_1) + \int K(x, y_2)\phi(x, y_2)dy_2 \tag{2.17}$$

Here, $\phi(x, y)$ is an arbitrary function with two domain variables x and y. $K(x, y)$ is a kernel function also with domains x and y. This type of equations have the same functions $\phi(x, y)$ on the left hand side and on the right hand side inside the integral. These integrals are called Fredholm integrals after Ivor Fredholm [Fredholm, 1903] and can not be solved analytically but with *Resolvent formalism* techniques where a discretisable property of the function can be represented as an infinite series (for instance, Liouville-Neumannn series) in order to approximate the solution of the equation. The two most popular way of solving the rendering equation are *Point sampling methods* and *Finite element methods*.

2.3.2.1 Point Sampling Methods:

Point sampling methods sample the entire domain to pick a finite number of points at random and evaluate the Fredholm Integral equations based on these individual values on each of these points. The number of points used to evaluate the integral is inversely proportional to the error. Although the sampling is a random process, there are many different sampling techniques which increase the efficiency of point sampling approaches by carefully picking the probability density function inside the domain of the function.

Section 2.4 describes the Monte Carlo approach [Metropolis and Ulam, 1949] of solving such integrals and also describes the most relevant sampling strategies. Section 2.5 will in turn describe the "Ray tracing" [Takagi, 1997; Whitted, 2005] and "Path tracing" [Lafortune and Willems, 1993] which are the point sampling methods to solve the rendering equation.

2.3.2.2 Finite Element Methods:

Unlike the point sampling methods, finite element methods are not dependent on random sampling. Instead, these methods divide the problem domain into smaller sub-domains with simpler solutions. Following the evaluation of the elemental equations of the sub-domains, the evaluated values are strategically recombined into a global system to put together the entire solution of the integral.

The next section will describe the Monte-Carlo method as a Point sampling technique to solve the rendering equation followed by the "Ray-tracing" and "Path-tracing" techniques. Section 2.5 will describe "Radiosity" [Cohen and Wallace, 2012] and "Instant Radiosity" as part of the finite element methods to solve the rendering equation.

2.4 Monte-Carlo Methods

The Monte Carlo approach [Metropolis and Ulam, 1949] to solve an integral of a function with respect to a given domain is to assign a random variable and pick samples from the given domain such that the expected value would be the approximate solution of the integral. Before describing the Monte Carlo integration and other related sampling methods in more detail, the following two sections will introduce the preliminary concepts of Random variables and Probability distributions respectively.

2.4.1 Random Variables

A sequence of random events, each having a subsequent numerical (in most cases) outcome and an associated probability of the outcome is collectively called a Monte Carlo sequence. In such a sequence, the summation of the probabilities of all possible values is 1. A variable **X** which takes a finite number of these outcomes is known as a *Random variable*. A discrete Random variable takes finite number of these outcomes while a continuous Random variable can take any number of possible outcomes often associated with a continuous function.

Expectation or the mean of a discreet Random variable **X** with n possible outcomes each having probability p is given by:

$$E[X] = \sum_{i=1}^{n} X_i p_i \qquad [\text{where } \sum_{i=1}^{n} p_i = 1] \tag{2.18}$$

Variance of the outcomes from the mean (expectation) of **X** is given by:

$$\sigma^2 = E[(X - E[X])^2] = \sum_{i=1}^{n} (X_i - E[X_i])^2 p_i \tag{2.19}$$

The variance is the measure of deviation from the mean or in case of Random variables, the measure of deviation from the Expectation $E[X]$. A process such as Monte Carlo which estimates the value of $E(X)$ has the goal of minimising the variance σ towards $\sigma \to 0$.

2.4.2 Probability Distributions

Probability Density Function (PDF), denoted as $p(X)$, can be defined over any arbitrary continuous random event so that the probability of a random outcome ($\mathbf{X} = x$) occurring is $p(X = x)dX$ where $\int p(X)dX = 1$. Here **X** is a continuous Random variable for the continuous random event over which the $p(x)$ is defined.

Cumulative Distribution Function (CDF) gives the probability of an outcome from a set of random events whose value is less than the given value **y**. The CDF for the outcome $[X = y]$, denoted as $P_{cdf}(y)$ is the probability of all possible outcome values that are less than equal to the value y ($p(X \leq y)$). The CDF of a continuous Random variable **X** is defined as:

$$P_{cdf}(y) = p(X \leq y) = \int_{-\infty}^{y} p(X)dX \tag{2.20}$$

Similarly for an interval:

$$p(a \leq X \leq b) = p(X \leq b) - p(X \leq a) = \int_a^b p(X)dX = P(b) - P(a) \tag{2.21}$$

Similar to the discreet Random variables, the Expectation and Variance for a continuous Random variable (X) are given by:

$$E[X] = \int Xp(X)dX \tag{2.22}$$

$$\sigma^2 = E[X^2] - E^2[X] = \int X^2 p(X)dx - (\int Xp(X)dx)^2 \tag{2.23}$$

2.4.3 Monte Carlo Integration

The goal of Monte Carlo technique [Metropolis and Ulam, 1949] is to estimate the value of an integral:

$$I = \int_a^b f(x)dx \tag{2.24}$$

where $f(x)$ is an arbitrary function defined over the domain $x \in (a, b)$. By definition, Monte Carlo process takes a finite number of samples of $f(x)$, $x \in (a, b)$. The selection of these samples is a random process and thus each sample carries a respective probability from a subsequent PDF $p(x)$. For a function G equaling to the sum of Random numbers g(x) weighted with w,

$$G(x) = \sum_i^N w_i g_i(x) \tag{2.25}$$

When the $w_i = w_{i+1}$ and $\sum w_i = 1$,

$$G(x) = \frac{1}{N} \sum_i^N g_i(x) \tag{2.26}$$

it can be proven that:

$$E[G(x)] = E[g(x)] \tag{2.27}$$

$$\sigma^2[G(x)] = \frac{1}{N}\sigma^2[g(x)] \tag{2.28}$$

Monte Carlo Estimator: As the expectations of G(x) and g(x) are same, G(x) can be used to estimate the E[g(x)]. Moreover, equation 2.28 shows that the variance of G(x) decreases as N increases, thus $\sqrt{N}$ is a factor for accuracy of the estimation of E[g(x)]. Tto solve equation 2.24, let us first assume that $f(x) = g(x)p(x)$ where

$p(x)$ is the probability of selecting the random variable $g(x)$. From equation 2.26 and 2.24,

$$G(x) = \frac{1}{N}\sum_{i}^{N}\frac{f(x)}{p(x)} = \langle I \rangle \tag{2.29}$$

where $\langle I \rangle$ is the *Monte Carlo estimator* of I. Any integral in the form of equation 2.24 estimated by Monte Carlo technique has its estimator in the form of $\langle I \rangle$.
It can be further shown that $E[\langle I \rangle] = I$ and the variance of the estimator $\langle I \rangle$ is:

$$\sigma^2 = \frac{1}{N}\int (\frac{f(x)}{p(x)} - I)^2 p(x) dx \tag{2.30}$$

This indicates that the variance σ^2 decreases as N increases monotonically. The standard deviation error is thus inversely proportional to $\sqrt{N}$. Thus, decreasing the error by half would require quadruple the number of samples. Solving an integral with Monte Carlo techniques consist of three steps; i.e. Taking N samples over a probability distribution, actually solving the function at that sample and then then take the mean of these solutions. Next sections will briefly introduce the different types of sampling procedures over a probability distribution.

2.4.4 Importance Sampling

Solving the integral $I = \int_a^b f(x)dx$ with the estimator $E[\langle I \rangle]$ requires the PDF $p(x)$ in accordance to which all the samples are drawn via any of the Inverse CDF, Cosine lobe or Rejection sampling methods. The optimal PDF $p(x)$ can be found by minimising the variance σ^2. it can be shown that $p(x) = \frac{1}{\sqrt{\lambda}}|f(x)|$ where λ is a scalar constant. This effectively shows that $p(x)$ will be optimal when it is in the shape of $f(x)$.

This points to two different cases to solve a Monte Carlo integration where either there is no information about the $f(x)$ i.e. *blind Monte Carlo*; and where there is some knowledge of the $f(x)$ also known as *informed Monte Carlo*. While there is no way to optimise $p(x)$ in case of blind Monte Carlo due to the absence of information about $f(x)$, the $p(x)$ can be shaped to resemble $f(x)$ in case of informed Monte Carlo with whatever information available on it.

2.4.5 Stratified Sampling

Clearly the Importance sampling does not work very well in case of blind Monte Carlo because of the often inefficient sample distributions. Moreover, even for an efficient PDF $p(x)$, the samples can be badly distributed inside the domain for a small number of samples resulting in "sample clumping". Although increasing the number

of samples will eventually mitigate this problem but it is significantly inefficient for faster, real-time solution requirements.

Stratified sampling solves the clamping problem by dividing the domain (Ω) into a finite number of "strata" ($\omega_0, \omega_1, ..., \omega_N$) and then evaluate each of the strata to find the solution of I.

$$\int_{\Omega} f(x)dx = \sum_{i=1}^{N} \int_{\omega_{i-1}}^{\omega_i} f(x)dx \tag{2.31}$$

It can be further shown that with same sized strata and one sample each stratum can make the Variance (σ^2) of stratified sampling less than simple, blind Monte Carlo. However, the N depends on the size of the strata compared to the size of the domain.

Monte Carlo integration and the relevant sampling strategies discussed in this chapter are crucial to solve the Fredholm integral [Fredholm, 1903] for light transport, i.e. the rendering equation that has been already discussed in Section 2.3. The next section will describe a few relevant Global Illumination (GI) algorithms which make use of these methods to solve the rendering equation and render photorealistic images.

2.5 Global Illumination Algorithms

The image of an object is formed when light bounces off its surface and reaches the camera. This incoming light into the surface of the object can come from either directly from a light source or reflected from other adjacent objects indirectly. In a complex, realistic scene, this indirect light transported via objects in the scene is equally important as the direct light for a realistic rendering. Global Illumination (GI) [Dutre et al., 2006] is a technique to address this transport problem of reflections and refractions of light inside a 3D scene while maintaining the energy equilibrium.

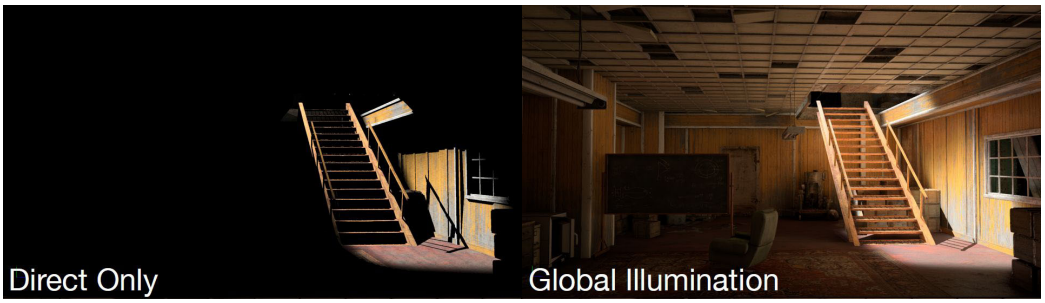


Figure 2.5: Comparison between Only Direct Illumination and Global Illumination with Indirect lighting. (Image courtesy: Mental Ray)

There are numerous GI algorithms that solve the rendering equation. As

described in section 2.3, the rendering equation assumes light is coming from the entire space around a point and it does not distinguish between whether it is a direct light source or an indirect one. The goal of a GI algorithm is to compute the expected radiance value on a point given a finite number of surface elements and the the solid angles between these surface elements and the said point. In order to do that, GI algorithms need to compute both direct and indirect illumination on a point.

GI algorithms can be classified into four groups. In terms of the strategy, these are either *Object-order* or *Image-order methods*; in terms of sampling, the rest two types of GI algorithms are either *Point-sampling methods* or *Finite element methods*. The *Object-order methods*, for instance the "Rasterisation" or "Radiosity" algorithms, iterates through different objects according to their order of occurrence in the scene. The *Image-order methods* go through each image pixel and find the object (or the environment) to calculate the colour value of that pixel. The most common of these algorithms are perhaps the Ray tracing based processes, such as Path tracing and its variants, Radiance Caching [Krivanek et al., 2005], Metropolis Light Transport [Veach and Guibas, 1997] etc. Ray tracing and path tracing family of algorithms are *point sampling methods*. In turn, there are finite element approaches for GI as well such as Radiosity. The following sections will discuss Ray Tracing, Path Tracing, Radiosity, and other relevant GI algorithms.

2.5.1 Ray Tracing

Ray tracing [Whitted, 2005, 1980] is one of the most frequently used image-order rendering techniques where a light ray between the light source(s) and the camera is traversed, calculating each interactions between the ray and the objects in the scene along its path. In the end, the resultant values are stored in the respective pixels on the image plane of the camera to produce a 2D image. To prevent computations of the light rays emitted from the light source which never reach the camera, the ray tracing works in the opposite direction, traversing from the camera to the light source(s). Figure 2.6 describes the Whitted-style ray tracing approach.

A ray tracing engine typically consists of three components; the *Ray generator* which generates a ray from any point to any direction required, the *Ray intersector* which checks the first object that intersects with the ray and detects the object properties of the hit-point. Whenever a ray hits an object, the final component, the *Ray shader* calculates the colour using the object properties of the hit-point.

The process starts from generating a ray from the camera towards a certain direction through one of the pixels in the image plane. This is termed as the "camera ray". When a camera ray hits a specular surface, the ray generator shoots a "reflected ray" from the hit-point. When a camera ray or a reflected ray hits a diffused surface,

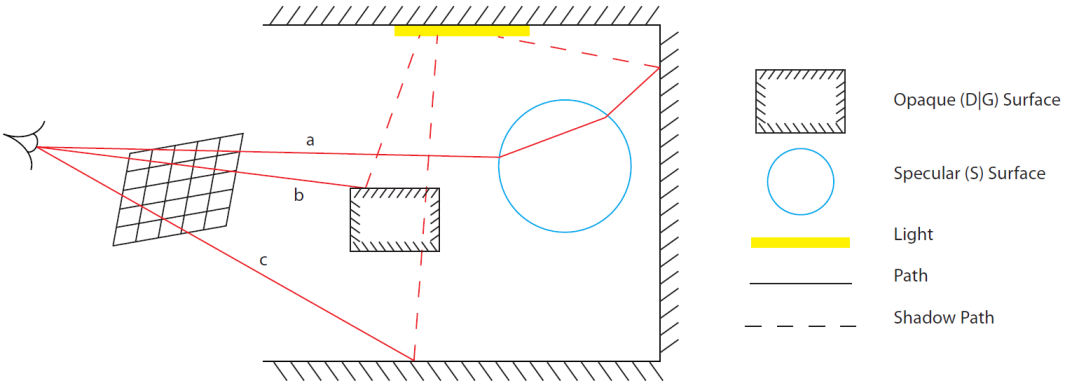


Figure 2.6: Whitted style ray tracing. Image courtesy: Bashford-Rogers et al.

the ray generator generates a "shadow ray" from the hit-point towards the light source. In the end, all the hit-points in that process are taken account by the shader to calculate the final colour of the pixel.

The biggest advantage of ray tracing is its simplicity. Given the powerful modern computers capable of processing hundreds of thousands of rays each second, ray tracing is used extensively for different types of renderings. However, it is not accurate because it does not take account the entire domain of the rendering equation. This gave the rise to other similar, more accurate approaches such as "Distributed ray-tracing" or derivative approaches like "path-tracing". The next section will discuss "path tracing" by Kajiya [1986] which takes the Whitted's ray tracing approach [Whitted, 2005] to solve the rendering equation much more accurately.

2.5.2 Path Tracing

The main disadvantage of ray tracing approach was the inaccuracy since it did not sample the entire upper hemisphere of a hit-point resulting in hard shadows and inability to render photorealistic phenomenons such as glossy reflection, depth of field, motion blur and umbra-penumbra regions. Path tracing improves it by employing Monte Carlo sampling to generate rays instead of just shooting shadow rays or reflected rays from each hit-points.

After a camera ray hits an object in the scene, the ray generator randomly generates a ray over a PDF according to the BRDF of the hit-point. The advantage of generating random rays over a PDF is speedup in case of partially specular and glossy surfaces. This random ray generation process is continued until either a ray hits a light source or the path reaches maximum length or "max depth". There are other strategies to terminate a path similar to this. The path is evaluated if it hits a light source.

Recalling equation 2.29 from the previous Section 2.4, a monte carlo integra-

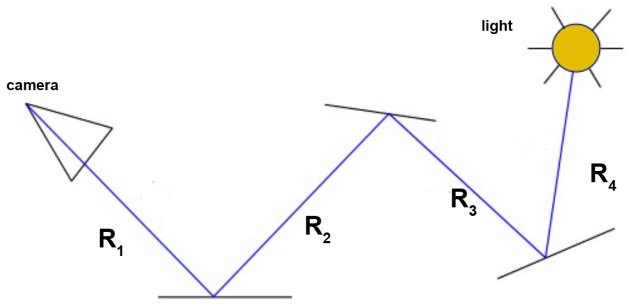


Figure 2.7: A simple path of rays R_1, R_2, R_3, R_4 hitting object surfaces in the scene finally connecting the camera with the light source. In path tracing, this while path from camera to the light source is calculated.

tion will require N different rays to estimate the incoming light (L) over the entire angular domain (Ω) at each hit-points. However shooting N number of rays at each hit points can be computationally very expensive and subject to "ray explosion" which is inefficient. Instead, path-tracing samples one ray per hit-point in the path and iterates this process for N number of time all the camera rays, averaging each iteration, producing a good estimate of the incoming light after sufficient such iterations. There are two advantages of this; firstly the algorithm produces a result, albeit inaccurate, after just one iteration so it is extremely fast and far more interactive than the distributed ray tracing approach of shooting N rays each point. Secondly, the improvement of results are seen in a logarithmic fashion as the number of iterations and different paths are added to the average pixel value.

Path tracing is one of the most popular among the GI algorithms due to its simplicity, accuracy, speed and easy implementability. However, the major drawback of path tracing type of algorithms is noise especially in the penumbra, soft shadow, colour bleeding regions on the scene. There are a few approaches that counter this problems by applying various intelligent albeit significantly more complex techniques to choose the paths that contributes the most for a given point in the scene such as "Metropolis Light Transport" (MLT) [Veach and Guibas, 1997].

2.5.3 Radiosity

While ray tracing and path tracing solves the rendering equation by *point sampling method*, "Radiosity" on the other hand uses *finite element method* to estimate the rendering equation. As mentioned previously, it is also an object-order method. This effectively means, unlike ray tracing and related methods, Radiosity does not have to recalculate and solve the rendering equation from scratch every time the camera position (view point) changes in the scene, thus making it "view-independent".

The basic idea of Radiosity is to divide the scene into a finite number of patches and calculates the transmitted lights between them, thus effectively simplifying the problem into solving a group of linear equations instead of the complex integral.

$$B_i = B_{ei} + \rho_i \sum_j F_{ij} B_j \tag{2.32}$$

Here, B_i is the Radiosity of the i^{th} patch in the scene. Similarly, B_j is the j^t patch in the scene whereas B_{ei} is the emitted radiance from the i^{th} patch and the ρ_i is the reflectivity of it. The new element here is the F_{ij} which is called the "form factor" between patch i (P_i) and patch j (P_j). The form factor depends on four geometric quantities; shape of P_i, P_j, relative orientation of P_i, P_j, distance between P_i, P_j and occlusion by any other patches in the middle. Depending on the complexity of scene geometry, calculating F_{ij} can be computationally expensive. A few techniques such as Hemicube (with z-buffer) [Cohen and Greenberg, 1985], Ray casting [Mora et al., 2002] and Analytical methods that calculates the form factor.

Depending on the accuracy requirements and scene complexity, it can be calculated reasonably fast to produce acceptable results and thus is extensively used in many applications. The view-independent nature is also one of the reasons of its popularity. However, the biggest disadvantage of Radiosity is that it assumes all surfaces as diffused, thus it cannot be used in many real-world scenes. Although there are a few Radiosity based algorithms that includes other BRDFs, they are computationally much more expensive and is unable to satisfy the primary purpose which is the computational turnaround time (speed).

This section described the relevant GI algorithms that has been extensibly used in the work done in this thesis, thus concluding the second set of sections describing "physical light transport". The following sections will describe various components of the relevant "*real-world light representation techniques*" including the state of the art such as Image Based Lighting, Lumigraph and Light Fields.

2.6 Lighting Techniques

As stated in the beginning of this chapter, the process of rendering consists of three components; i.e. 1) scene geometry, 2) lights and 3) a rendering algorithm that calculates the light transport in the scene and outputs an image. In computer graphics, these lights can be *virtual* or *real-world*.
Virtual lights are virtually described geometries (point, area or volume) that emit light. They can be easily represented, managed and manipulated. It is easy to create simple and consistent virtual light sources. However, the disadvantage of virtual

lights are its limitations in representing the complexities of lights in the real-world. *Real-world lights*, on the other hand, are hard to capture and implement in a synthetic scene. Depending on the requirements, it can be difficult to represent, manage and manipulate. Their main advantage is the accuracy which is often crucial for photorealism [Lyu and Farid, 2005].

The following sections will describe the "plenoptic function" used to represent light, the high-dynamic range (HDR) imaging and the ways to capture and use it for various Physically-Based rendering techniques.

2.6.1 The Plenoptic Function

Real-world lights are often based on high-fidelity HDR images of a real world scene including the primary light sources such as the sun. The image based methods described in the following Sections 2.7, 2.8, 2.9 and 2.9.4 using real-world light includes Image Based Lighting [Debevec, 2002], Lumigraph [Gortler et al., 1996], Light fields [Levoy and Hanrahan, 1996] and Dynamically parameterised light fields [Isaksen et al., 2000] respectively. The main objective of such family of methods is to capture light rays propagating in the scene and later reconstruct the scene from it. The plenoptic function is a viewpoint independent description which helps representing and later reconstructing light rays in a 3-D space. Figure 2.8 provides a visual depiction of the it.

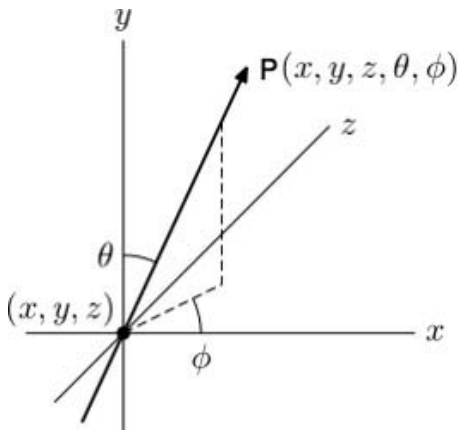


Figure 2.8: Components of the plenoptic function

The "plenoptic function" (P) describes the radiance (or chromacity) of light rays propagating through a 3-D space. P is defined by $P(\mathbf{x}, \vec{\omega}, t, \lambda)$; a function of origin position $\mathbf{x}$, angular direction $\vec{\omega}$, time t and wavelength λ [Adelson and Bergen, 1991]. This is further reduced to $P(\mathbf{x}, \vec{\omega})$ with constant time and wavelength turning it into a 5D plenoptic function with three dimensions (X, Y, Z) for ray origin co ordinates

and two dimensions (θ, ϕ) for angular direction.

$$P = P(x, y, z, \theta, \phi, t, \lambda) \tag{2.33}$$

$$P = P(x, y, z, \theta, \phi) \qquad \text{[Reducing dimensionality]} \tag{2.34}$$

The plenoptic function is measured at only one point in the case of traditional Image Based Lighting (IBL) described in section 2.7 later, and at every point on the rendering space in case of various types of Light Fields (LF) and Lumigraphs described later in section 2.8, 2.9 and 2.9.4 respectively. In all cases, HDR images of the environment can be deconstructed into a plenoptic form (as shown in Equation 2.34) describing the light. The reconstruction of the scene from the light description is done by finding the luminance in a particular point x through a certain direction ω.

2.6.2 High Dynamic Range Imaging

If a scene has light sources and shadow parts, it is often often seen that an image of the scene is either very dark underexposed shadow parts and properly exposed light sources or completely overexposed light sources with illuminated shadow parts. Both of these images do not carry the entire range of luminance values in the scene and information is lost from both extremes. A High Dynamic Range(HDR) [Banterle et al., 2017] image is an image which covers a very large portion of the dynamic range of a given scene; enabling flexibility and providing enough information to render from such images treating the image of the light sources as actual sources of light. A typical HDR image has enough information in the shadow region as well as distinct edges of the light sources.

The **Dynamic Range** (DR) of a camera sensor is defined as the ratio between the highest detectable illuminance which does not saturate the sensor and the lowest detectable signal [Mukherjee et al., 2016]. DR is expressed in terms of *f-number* also known as *f-stop* or *relative aperture* which is a dimensionless quantity. A three f-stop image has double dynamic range of a two f-stop image. Prior to the digital imaging era, the film provided a DR of approximately 14 f-stops. Modern state of the art digital sensors with 14-15 f-stops can surpass the dynamic range of films, although it is still not enough to capture and represent real-world lighting accurately. For this reason three or more still images are taken of the same scene with varying exposures and merged together to form a single HDR image.

While this merging method works for still images, capturing HDR videos are much harder because the object is moving rapidly from frame to frame and there is not enough scope for multiple exposure images in that small time. The RED Epic and ARRI Alexa, two professional video cameras attempts to somewhat solve this with a

high dynamic range sensor capable of capturing videos consisting of individual frames of upto 14 f-stops in a very high resolution and frame speed. However the exclusivity of these cameras has always been a restricting factor in HDR image based rendering technique. There are a few approaches [McNamee et al., 2015], [Unger et al., 2004] that attempts to solve this problem with consumer grade hardware which makes HDR imagery more accessible for photorealistic rendering.

2.6.3 Capturing Real-World Light

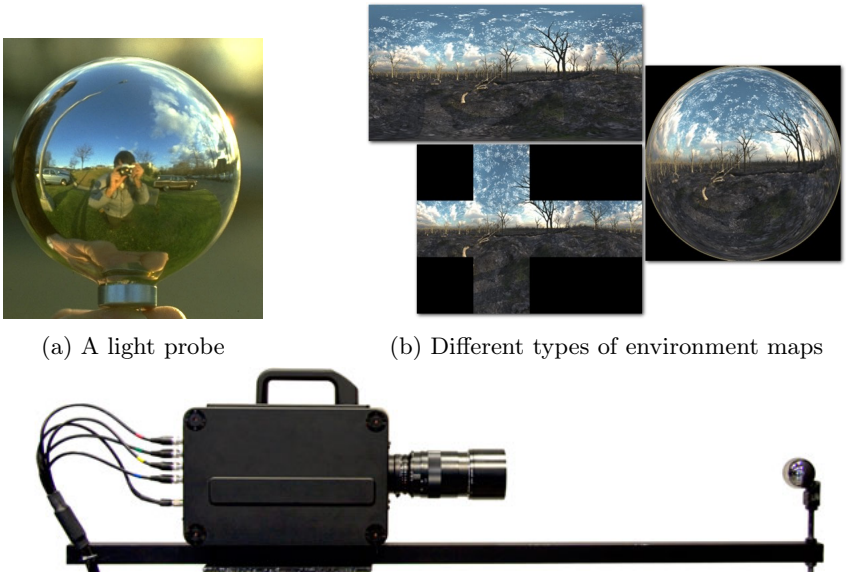
There are two main uses for real images in a renderer; as *texture* to an object and as an *environment map* for lighting. Capturing an environment map is done by capturing a *light probe*; a 360° panoramic image of the upper hemisphere of the scene's surrounding (environment). There are different ways of capturing such an image. The most common technique uses a highly reflective and precisely spherical mirrored probe (figure 2.9a) which reflects the environment. An HDR camera is used to take a picture of the probe capturing the entire ($360° \times 180°$) angular domain of the hemispherical environment.

Depending on the requirements, the environment map can be of different types other than the circular polar images. Figure 2.9b shows the latitude-longitude and the cubic maps. The number of such environment maps depends on the type of light representation technique is used. For example, Debevec et al. used single HDR light probes from the spatial point of the synthetic object in the real scene[Debevec, 2008] in Image Based Lighting. On the other hand, complex techniques such as Incident Light Fields use multiple environment maps across the volume of the scene.

This section described the means of capturing real-life light to render a virtual scene. The following sections will describe the most widely used relevant techniques such as Image Based Lighting, Lumigraph, Light Fields and Dynamically Reparameterised Light Fields, which uses images of a real scene.

2.7 Image-Based Lighting

One of the frequent problems in computer graphics is to render a synthetic object in a real-world scene. Image Based Lighting (IBL) is a way of illuminating virtual objects with images of the natural environment itself [Debevec, 2002]. The central idea of IBL stemmed from the texture mapping technique [Blinn and Newell, 1976]. The process starts by capturing a High Dynamic Range(HDR) environment map of the real scene. The illumination data from the environment map is considered as the primary light source in the various global illumination algorithms to be used for the final rendering. In the end, the properly illuminated virtual object is merged with a real picture of the scene.



(a) A light probe

(b) Different types of environment maps

Figure 2.9: An HDR Light Probe. Left is an HDR camera and at the extreme right, the silvered ball can be seen capturing the 360° panoramic image of the environment.

IBL is a reasonably fast and efficient technique for photorealistic rendering. It has been used commercially for various CGI works in the mainstream movie industry to great extent for its implementability. This section will describe the main methodology of IBL.

Figure 2.10: Synthetic objects illuminated with real-world light with IBL. (Image courtesy: Debevec et al.)

2.7.1 Methodology

Believability of a photorealistic rendering depends on accurate illumination of the synthetic object according to the illumination of the scene. Before IBL, this was simulated by synthetic light sources but this approach had a huge drawback because determining the exact positions and radiometric properties of the light sources in the real scene manually is a tedious and an error prone process. Debevec et al. used the HDR light probe image as the primary source of illumination and achieved high-fidelity rendering of synthetic objects inside a real scene (figure 2.10) [Debevec, 2008].

2.7.1.1 Scene Segmentation

The IBL pipeline is divided into three parts according to the mutual distance of the scene: the distant scene rendering, local scene rendering and the synthetic object rendering.

The distant scene is defined as the area of a scene where no part of it is being affected by the light rays reflecting back from the synthetic objects. The distant scene is rendered out of high resolution and high dynamic range real-world images. The distant scene also serves as the primary illuminant for the synthetic object.

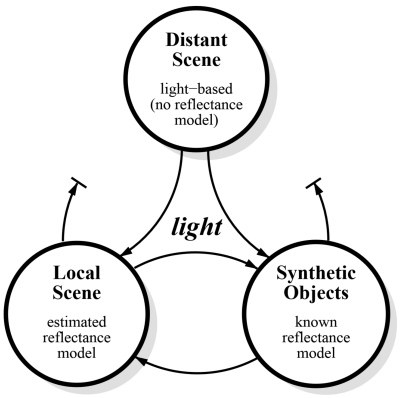


Figure 2.11: Scene segmentation in IBL rendering pipeline. Image courtesy Debevec et al.

The local scene on the other hand is fuzzily determined. Often the scene area nearest to the object is the local scene. Sometimes however, depending on the material and reflective property of both the object and adjacent scene, light can concentrate to some part of the scene, significantly affecting the luminance of it. Lastly, **the synthetic object** part is the primary rendering part where the object is illuminated using the environment map given the BRDF of it.

2.7.1.2 Differential Rendering

The rendering workflow starts with capturing the HDR light probes and obtaining a detailed model of the synthetic object to be rendered along with its BRDF data to accurately simulate the reflection/refraction via the global illumination algorithms. The rendering is done in three parts mentioned before, distant and local scenes and the synthetic object. The first part(LS_b) is the light based model, i.e. the background image. Second part is the local scene without the object(LS_{nob}) while the third part is the rendering of the local environment with the object(LS_{obj}) calculated by the global illumination algorithms. The final rendering (LS_{final}) is computed as:

$$LS_{final} = LS_b + (LS_{obj} - LS_{nob}) \tag{2.35}$$

2.7.2 Temporal IBL

The gradual adaptation of HDR video has seen approaches which takes the IBL to the temporal dimension by recording the light probe image with a HDR video camera, resulting in frame by frame representation of the scene. These frames are then processed to obtain a temporal coherence with which the synthetic objects are relighted.

There are various methods that takes the proven principles applied to the IBL to produce temporal IBL techniques. One such method which is relevant to the work in this thesis is the temporal filtering by Havran et al [Havran et al., 2005]. The approach detects the light source in the environment map and filter their power and position across frames which avoids recomputing the whole scene for each frame.

There are many other methods such as sequential sampling method [Ghosh et al., 2006] and light clustering method [Sato et al., 1999] are there most of which implements different approaches to isolate the light sources and change the importance sampling filter throughout the temporal space. These approaches, however different from the current work, provides a background to the state-of-the-art in adding the temporality while capturing and rendering from real-world light.

2.7.3 Limitations of IBL

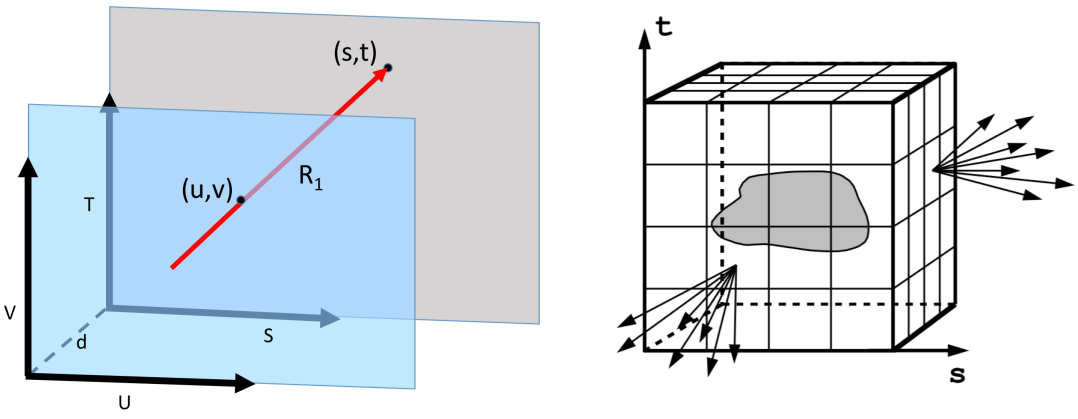
Although IBL is an efficient photorealistic rendering system, it sometimes produces rendering errors due to the difference (Err_{ls}) between the local and the distant scene ($LS_{nob} - LS_b$). This also causes the rendered image to have parallax errors while rendering a sufficiently large scene and not just a single object. These limitations has inspired other variants of IBL. The work described in this thesis is closely related to such a "Light Field" based variant of IBL, the *Incident Light Field (ILF)* which will be described in detail in Chapter 4.

2.8 Lumigraph

The Lumigraph technique captures the "complete appearance" of a scene or an object and represents it in an efficient plenoptic representation and eventually reconstructs the object or the scene from arbitrary camera viewpoints; with added functionalities such as depth data and in some cases, changing the lighting in the scene [Masselus et al., 2003]. The Lumigraph is a 4D plenoptic function of the light around a bound object.

2.8.1 The 2-plane "uv-st" parameterisation

There are four most popular parameterisation of a ray (of light) depending on the representation used[Camahort et al., 2009], [Levoy, 2006]. The "2-plane" parameterisation, more commonly known as "uv-st" parameterisation uses two 2-dimensional planes, respectively the u-v plane and the s-t plane to represent the origin and the direction of a ray. It is not necessary for the planes to be parallel and in many light field applications it is often at different angles to each other. However for the scope of Lumigraph, UV and ST are assumed to have constant distance between them. Figure 2.12a shows the two planes UV (blue) and ST (grey) parallel to each other at a distance (d). The ray R_1 is intersecting UV plane at (u,v) and the ST plane at (s,t). Together the ray can be represented as (u,v,s,t) as shown in the Figure 2.12a.



(a) The 2 Plane UVST parameterisation

(b) A Lumigraph cube around an object

Figure 2.12: The Lumigraph and its 2-plane parameterisation

2.8.2 Discretisation

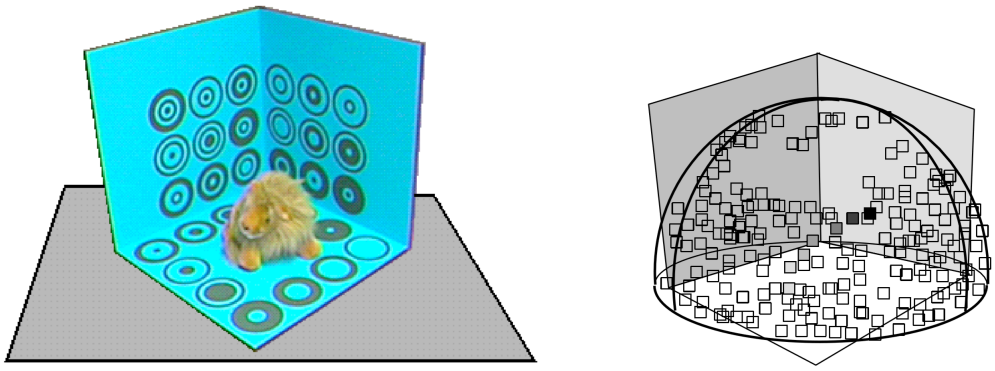
The Lumigraph is a 4 dimensional continuous function where the plenoptic function is represented on the surface of a 3-D cube surrounding an object or a scene (Figure 2.12b). For ease of computation, the plenoptic function is discretised in

the Lumigraph. There are M subdivisions indexed as (i,j) for the (s,t) plane and N subdivisions indexed as (p,q) for the (u,v) plane. The data value at the each grid point is referred as $x_{i,j,p,q}$. A basis function $B_{i,j,p,q}$ is associated to reconstruct the continuous Lumigraph (L) as a linear sum. The choice of the basis function can be a constant as unity or it can accept quadrilinear basis function where at the grid point the value is 1 and falls to 0 at all neighboring points.

$$x_{i,j,p,q} =< L, B_{i,j,p,q} >$$

2.8.3 Capture

The camera is a virtual pin hole camera placed at each grid point (i,j) centered at the point $(s_i.t_j)$ focused at the hyper-focal distance and looking straight along the z-axis. The pixel values in the images are used as he values $x_{i,j,p,q}$. The capturing system of the Lumigraph is a regular camera with precise calibration mechanism. The capture positions respect to the object is calculated by the special arrangement of the "capture stage" and a few "markers". The capture stage is built with two cyan coloured walls joined with each other at 90°s. The base is also a cyan coloured square which can be rotated in 90°s increment. The walls and the base have numerous concentric circles in deeper shade of cyan to act as markers (Figure 2.13a). At a time 8 or more markers need to be visible for each frame of the Lumigraph. A lot of these images are to be taken in variable sampling rates throughout the stage to capture the Lumigraph (Figure 2.13b).



(a) Capture stage.

(b) Camera positions of the captures on the viewing sphere.

Figure 2.13: Capturing a Lumigraph. Image Courtesy Gortler et al.

2.8.4 Limitations

Lumigraph technique is highly effective for capturing the entire view of objects and scenes which later can be reconstructed from arbitrary camera position and even

approximate the 3D volumetric shame of the object by capturing picture samples around the object/scene on a specially constructed capture stage. Unfortunately, this remains Lumigraph's biggest limitation as well.

Identifying the capture points in a large 3D scene is not trivial and poses many real-world difficulties. Lumigraph's usage of the capture stage is an effective way of capture point identification but this makes it unviable for large objects or real-world scenes.

A very similar approach to Lumigraph was taken by Hanrahan et al. which represents outgoing light rays from an object or a scene into a 4 dimensional field [Levoy and Hanrahan, 1996]. The next section will describe the "light field" technique to represent and later reconstruct a real-world object or a scene.

2.9 Light Fields

Since the texture mapping technique [Blinn and Newell, 1976] there has been efforts for making realistic image based renderings and reconstructions with arbitrary views of real environment. One of the first such approaches warped 2D images of real scenes to give the rendered scene the perception of depth and projection [Chen, 1995] independent of the view angle. Apple incorporated this technique in their *Quicktime VR* software in the early nineties.

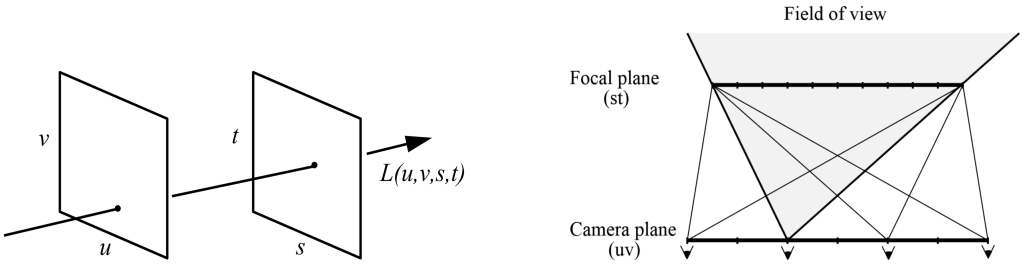
The major drawback of these methods was their viewpoints were fixed. Although it was possible to interpolate the images to fit the new viewpoints, the procedure required depth data which was difficult to provide from 2D images of the environment. The introduction of Lumigraph [Gortler et al., 1996] described previously, provided ways to reconstruct scenes and objects from arbitrary view points. The plenoptic function based parameterisation of captured light rays enabled Lumigraph to represent, store and reconstruct efficiently, albeit with a complicated capturing method.

Light Field [Levoy and Hanrahan, 1996] captures its samples along a camera plane rather than samples around the subject like Lumigraph.

2.9.1 Representation

The Light Field is defined as the amount of radiance from a particular point towards a particular direction. It is represented by a 4 Dimensional plenoptic function (given the sampling is in 2D surface) P having 4 parameters defining the positions of the point(Ori_x, Ori_y) and the direction(Dir_x, Dir_y) of the ray.

$$P = P(Ori_x, Ori_y, Dir_x, Dir_y)$$



(a) Light slab representation

(b) Viewing geometry to create a light slab

Figure 2.14: The parameterisation and schematic of the light slab generation in Light Fields. Image courtesy Hanrahan et al.

The parameterisation of the light ray space is based on a light slab representation (figure 2.14a) where there are two planes; one for origin of the ray and one serves as the exit plane of the ray. The cartesian co-ordinates of the two planes (s,t and u,v) are used as the 4 parameters to represent the plenoptic function.The light slab representation is good for efficient calculation and it also enables uniform sampling of the images of the scene. Figure 2.14b depicts the sampling row (at the camera plane) of a scene (at the focal point) to create a light slab out of the sampled images.

2.9.2 Methodology

The light field is constructed from a number of 2D images of the scene to be rendered from varying points in a plane (camera plane). Each point of the images represents one light ray origin and the its direction is determined by the relative position of the point in the image and the same point in the actual scene. The image plane here serves as the entry plane and the focal plane of the images serves as the exit plane. The rays can be easily constructed from these two planes and a collection of these rays can construct a *light slab* with illuminance data for the actual rendering.

After creating the light slab, it is necessary to store it after a compression procedure to (1) get rid of data redundancy, (2) enable easier random access to the ray database and (3) making it computationally less expensive to store and retrieve data during rendering. The light slab data is compressed in two steps. A lossy vector quantization where decoding the compressed data is very fast because of reproduction vector called codewords and efficient organization of a set of codewords into codebooks. The second step is to employ an entropy coding like Lempel-Ziv [Ziv and Lempel, 1978] to compress high probability codeword indices.

Reconstructing images from the Light Field is the final step of the entire rendering process. It is done in two steps: (1) calculating the u, v and s, t parameters to construct the ray; (2) reconstruct the radiance sample from the ray. Given the entry

and the exit plane, it is quite trivial to compute the (u,v,s,t) parameters from the image co-ordinate (x,y) as a projective map via texture mapping. The re-sampling of the radiance procedure first constructs the function from the original samples and then applies a bandpass filter to reduce any aliasing effect. The re-sampling process is approximated by either a nearest neighbor or a bilinear interpolation.

The entire process of the Light Field started a new direction in image based models where it supported arbitrary camera positions for the first time in the history of realistic rendering. The Incident Light Field has also has been developed on a similar technology where the rays are constructed from images of the scene, compressed, stored into light slabs and re rendered efficiently. There is a very similar technology to Light Field that was there independently at the same time. We will now briefly describe The Lumigraph in the next section.

2.9.3 Limitations

Despite its many capabilities and prospects, Light Fields have a few limitations.

1. Static focal plane: The focal plane of LF is fixed and thus can only be focused on a single plane. If an object spans considerable are, the entire object can not be focused at once.

2. Restrictions of the aperture: The image plane in the LF is fixed and the apertures are fixed at the capture time depending on the camera positions and image density. These can not be modified afterwards.

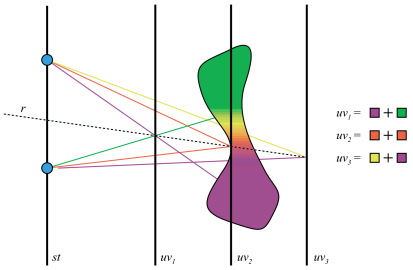
3. Can't render synthetic scenes: LF only reconstructs the captured scene. It can not be used directly to render synthetic objects with the light available in the captured scene.

In order to mitigate these limitations, there are two prominent LF based technologies that offers much more flexibility into LF. The first one is Dynamically reparameterisation of LF which is discussed in the following section. The other technology is the Incident Light Field (ILF) which has been described in detail in the next chapter since the work described in this thesis has been based on ILF, eventually augmenting it to develop the novel Temporal ILF.

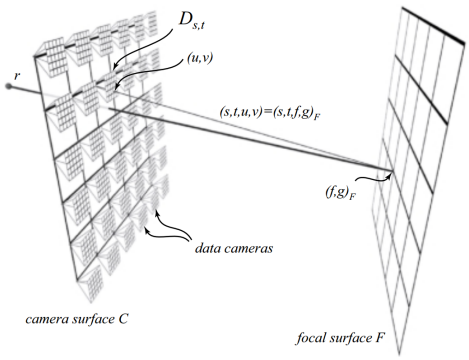
2.9.4 Dynamically Re-Parameterised Light Fields

A number of approaches involving the Light Field technique have been made which addresses the parameterisation limitations of the Light Field. One of the prominent among these is the dynamically re-parameterised light field Isaksen et al. [2000] which can have multiple focal planes and an arbitrary aperture during reconstructing a Light Field. In a similar approach, a new type of camera has been created where

different objects can be focussed *after* taking a picture Ng et al. [2005]. Although these approaches are not directly related to render synthetic scenes, their unique approaches to achieve flexibility in rendering light fields are relevant to the work presented in this thesis.



(a) Focal Plane Parameterisation - (u,v) planes are the exit planes



(b) Camera array - a collection of data cameras $D_{s,t}$, and a dynamic focal surface F. Each ray (s, t, u, v) intersects the focal surface F at $(f, g)_F$ and is therefore named $(s, t, f, g)_F$

(c) Seeing through obstacles in high synthetic aperture

Figure 2.15: Dynamically re-parameterised Light Fields. Image courtesy: Issacsen et al.

While the LF technology is based on building a ray database from multiple images on a row, Isaksen et al. Isaksen et al. [2000] took the technology further to build a ray database which is moderately sampled but with very high depth information and a parameterisation scheme where specific rays can be selected from the database to mimic arbitrary synthetic aperture. The main goal of the work is to have wide variation in depth without requiring extra geometry information from the scene (unlike Lumigraph). The work employs focal plane parameterisation to select light rays from the ray database that focuses on a specific focal plane. This way any arbitrary aperture can be mimicked. An application of this is seeing through any obstacles with a very high synthetic aperture (Figure 2.15a).

In order to build the ray database, Isaksen et al. built a camera array con-

sisting of many cameras in a grid (Figure 2.15b). Besides the arbitrarily large synthetic aperture, the dynamic parameterisation algorithm enables a picture to be auto-stereoscopic as well.

2.9.5 Lytro: The Light Field Camera

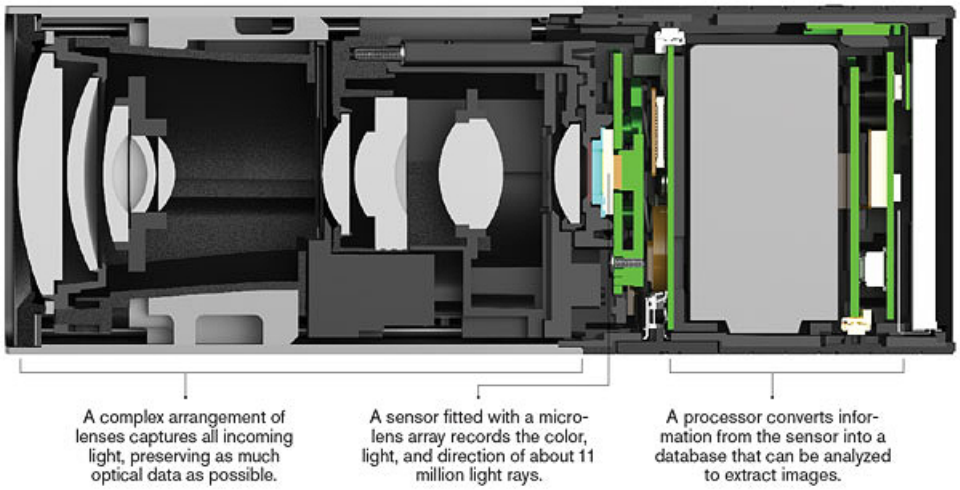


Figure 2.16: Lytro: Light Field Camera cross section. Courtesy: Lytro

The re-parameterised Light Field was refined into a more scalable yet compact solution with the Light Field Camera named Lytro. Ng et al. Ng et al. [2005] developed a technique using a micro lens array instead of the camera grid used by Isaksen et al. in front of the camera sensor to get multiple tiny but spatially variant images with different focal plane. Figure 2.16 shows a disection of a prototype lytro camera with different components labeled.

The images from the micro-lens array are combined to make a ray database of the entire scene similar to the other LF techniques. Similar to the previous approach, rays representing a specific focal planes are selected from the multiple images. The ray database can be manipulated in the frequency domain to refocus images after the image has been taken.

Although the above two technologies are quite different from the illumination models such as IBL, these unique uses of the plenoptic function and a ray database helps to understand the versatility of the Light Field technology. The next section will revisit the background provided in this chapter and discuss the evolution of a spatially variant image-based object illuminantion technique called the Incident Light Field (ILF)[Unger, 2009] which has been described in detail in the next chapter.

2.10 Discussion

This chapter has provided the relevant background for the photorealistic rendering technique. Rendering requires scene geometry, light information and the light transport algorithms. This chapter discussed various physically-based quantities and then the physically-based light transport techniques in detail. There are several approaches that has their own advantages and drawbacks and are employed on different purposes.

In addition, this chapter describes the state of the art techniques for real-life representation techniques which can be realised with the plenoptic function. IBL, Lumigraph and LF are such representations having specific advantages and respective drawbacks. While the IBL can not represent spatially varying light information, LF has been designed for representing spatially varying light. On the other hand, IBL is extensibly used for rendering synthetic scenes with the light available in the environment while LF has been developed for reconstructing a scene with added flexibility and functionality and not suitable for rendering synthetic objects.

Incident Light Field (ILF) [Unger, 2009; Unger et al., 2003] is a technique that combines IBL and LF to provide a solution to represent spatially variant lighting for rendering synthetic objects lit with real-world light. Chapter 4 will describe ILF, its implementation and its primary limitation in detail.

ILF is an important part of the work done in this thesis. The DCP technique and the Temporal ILF technique developed in this thesis is interconnected with the ILF technique. Thus, prior to describing ILF in Chapter 4, the next chapter will elucidate the interconnection between the various parts of the work done in this thesis.

Chapter 3

Methodology

"People know what they do; frequently they know why they do what they do; but what they don't know is what what they do does."

– **Michel Foucault**

This brief chapter provides a holistic, concise visualisation of the overview of the work done in this thesis. The primary objective of this thesis, as stated in Chapter 1, is to develop the "Temporal ILF"; a novel technique which can capture and represent real-world spatio-temporally varying light.

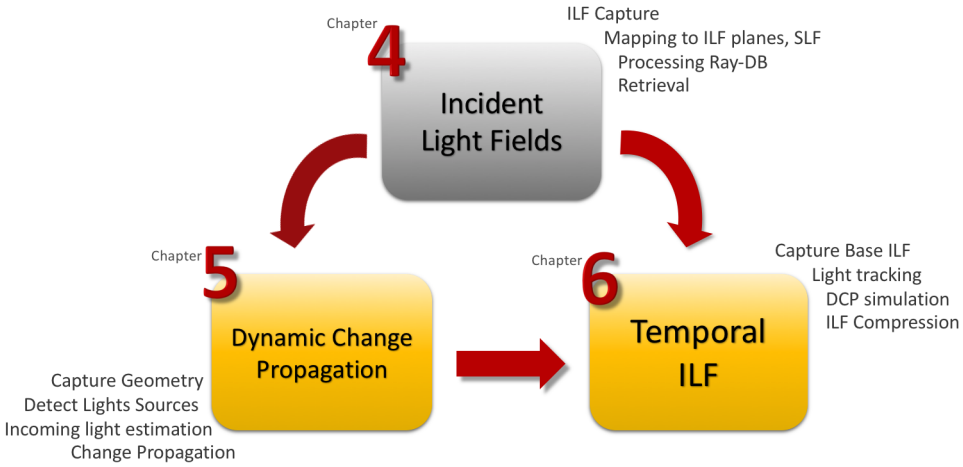


Figure 3.1: A schematic diagram of the interconnections between Incident Light Fields (ILF), Dynamic Change Propagation (DCP), and Temporal ILF with the subtexts around each boxes reflecting the basic parts of their individual pipelines. Chapter 4, 5, and 6 of this thesis describes each of these techniques respectively.

As Chapter 1 has discussed, Temporal ILF cannot be captured by acquiring temporal snapshots of static ILF captures in the scene because a single static

ILF capture is not instantaneous. This is why the Temporal ILF is captured in two steps: (i) An initial static ILF capture and (ii) Tracking the position and radiometric properties of the light sources in the scene. After this two step capture, the changes recorded in the light source tracking is then synthetically implemented on the initial static ILF to obtain any number of synthetically changed ILF snapshots, thus eliminating the need of physically capturing individual static ILF for each change in the light sources over time.

After the capture, the synthetically editing the static ILF is done by a novel technique called the "Dynamic Change Propagation" (DCP) which not only change the light sources synthetically in the static ILF but also calculates the effects of these changes in the indirect lighting stored in the ILF and updates it accordingly. Figure 3.1 shows the interconnections between the three different technologies. In the diagram, ILF is a part of the DCP, and both ILF and DCP are two primary parts of the Temporal ILF.

The implementation of the ILF technique [Unger, 2009] in this work, described in Chapter 4, has some representational changes from the original literature for free-form ILF [Unger et al., 2008] to aid the specific requirements of DCP and the Temporal ILF while keeping the general principle intact. The DCP technique has been developed to aid the Temporal ILF, however, it can also be implemented on its own for a number of applications which has been discussed in the Chapter 5. These two techniques together forms the Temporal ILF described in Chapter 6.

The following sections will describe the primary principle that guides the process towards the development of the Temporal ILF; as well as, go through the overview of the general methodology of each of these three techniques (ILF, DCP, and Temporal ILF) before the following Chapters 4, 5, and 6 describes them respectively in greater detail.

3.1 Primary Principle

The objective of this work is to develop a technique capable of capturing real-world light which varies spatially (Figure 1.3) and temporally. As a single light probe image is not enough to represent the spatial variations, the Image Based Lighting (IBL) technique [Debevec, 2002], which assumes a constant and uniform light throughout the rendering space is inadequate for a complex realistic lighting scenario. A related technique is the Temporal IBL [Havran et al., 2005] which captures light at a single point over time. As this too lacks the spatial variation, Temporal IBL is not an effective solution for more complex lighting situations.

As shown in the flowchart in Figure 3.2, the ILF technique [Unger, 2009] is capable of capturing 5 dimensional complex lighting ($P(x, y, z, \theta, \phi)$) by capturing a

dense spatial sampling (often thousands per unit square area) of light probes throughout the rendering space. The six dimensional Temporal ILF ($P(x, y, z, \theta, \phi, t)$) can be captured by acquiring multiple snapshots of the ILF of a scene over time.

However, as it is not practically feasible to have a camera at every possible point in the rendering space, a dense spatial sampling is achieved by moving an HDR camera along this space. This capturing method is not as instantaneous as taking a single light probe and can take a reasonable amount of time depending on the area of the capture. Thus, capturing multiple snapshots of the ILF over time to capture a Temporal ILF this way would be infeasible, especially in scenarios where the changes in the light occur rapidly and arbitrarily over time. The flowchart in Figure 3.2 shows this problem in capturing multiple snapshots of the ILF.

A solution to this infeasibility of physically recording multiple snapshots of ILF over time is to synthetically simulate the changes in the light sources over time in a captured static ILF. The ILF of a static scene is captured once with all the light sources in the scene. Subsequently, the changes in the light sources are tracked separately in order to extract the position, colour and intensity of each of the light sources with a reasonably high temporal resolution (24 FPS for movies, 30-60 FPS for computer games applications). Later these changes are synthetically simulated to the static ILF by editing the light sources in the ILF ray-database accordingly.

The primary problem with this approach is, changing solely the light sources do not change the indirect light information already recorded in the original ILF. In real-world, changes to the direct light sources changes the indirect light. The "Dynamic Change Propagation" (DCP) technique developed in this work estimates the effects of any synthetic changes made in the direct light sources and propagates them to the entire indirect lighting information in the ILF database.

As shown in the flowchart in Figure 3.2, the DCP technique makes it possible to simulate multiple snapshots of a static ILF. The Temporal ILF technique is recorded by capturing the base ILF of the static scene once in the beginning and subsequently recording all the positional and radiometric changes made to the light sources. The DCP technique is then used on the static ILF to simulate these recorded changes in the light sources to obtain simultaneously spatially and temporally varying light representation of any given real environment with indirect light fidelity.

The following sections will briefly discuss the synopsis of the respective works to provide a preliminary understanding of the different parts of the work done and how they are interconnected.

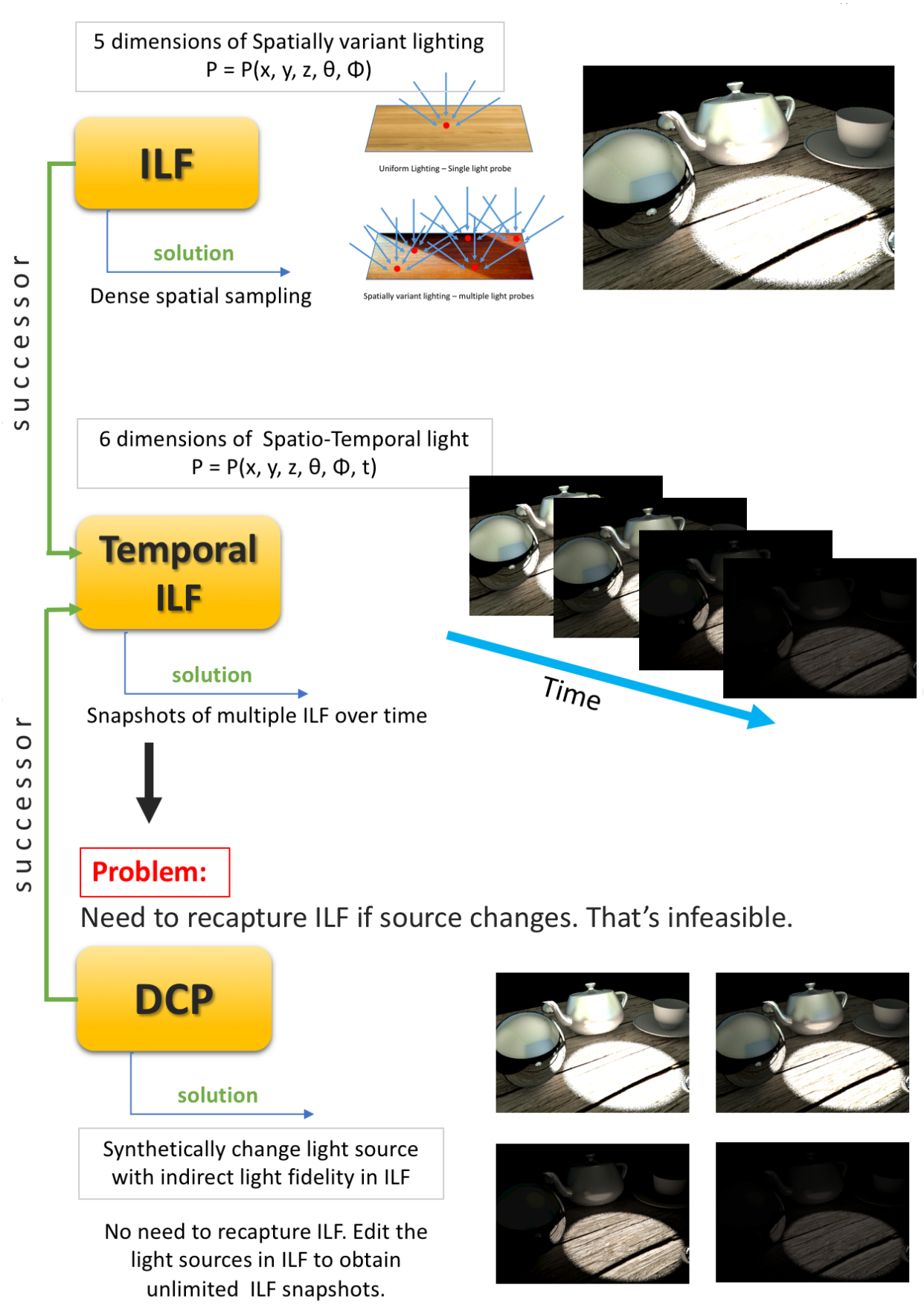


Figure 3.2: A problem-solution-result flowchart.

3.2 Incident Light Fields (ILF)

ILF, introduced briefly in Chapter 1, is one of the best techniques for representing spatially variant lighting for rendering a synthetic object. ILF is a special type of Light Field where instead of capturing only a part of a scene, the whole environment is captured with multiple light probes.

3.2.1 Representation

The Light Fields (LF), as described in Chapter 2, produce a dense ray-database of a scene by capturing a number of images of the same scene from different spatial position of an imaginary 2D plane called the "Image plane". Each pixel of these images represent a light ray coming from the scene to the specific capture point in the image plane along its specific direction. These "rays" with a position and a direction (x, ω) carrying the light information are agglomerated together as a ray-database.

ILF uses the similar technique to construct a ray database for incident light which comes from the entire environment into the rendering space. However, instead of using planar images like LF, ILF uses light probe images to capture the entire range of incident light from the upper hemisphere of the rendering plane.

3.2.2 Implementation

There are a few different ways to represent an ILF which has been discussed in Chapter 4. One of the most versatile techniques is to map the ray database to the surfaces of a bounding box encompassing the rendering space. This is explained in greater detail in the respective Chapter 4. The dimensions of this bounding box is arbitrary with the only condition being to encompass the entire capture (and rendering) space.

This work implements ILF slightly differently. Instead of mapping the ray-database into the arbitrary bounding box, it is mapped to the actual geometry of the scene. Once again, Chapter 4 explains the method in much greater detail. This difference in the mapping does neither improve nor decrease the quality of ILF renderings in any way, however, greatly aids to the implementation of the DCP technique which estimates the lighting information whenever a change is made to the light sources in the ILF database. DCP is briefly introduced in the next section.

3.3 Dynamic Change Propagation (DCP)

The ILF ray-database can store the direct light coming from the light sources and the indirect light reflected from the background geometry of the scene separately.

Separating the direct light sources from the indirect light coming form the background greatly reduces the rendering time with optimised renderers [Unger, 2009]. Typically, the direct light sources in the ILF can be altered synthetically in terms of intensity and colour. However in the real-world, any changes to the direct light sources should also change the indirect light reflected from the scene background. ILF on the other hand can't calculate the effects of the synthetic changes made in the light sources to the indirect light because it has no information about the light the various part of the scene receives. DCP is the developed novel technique which estimates the effects of any arbitrary changes in the light sources and propagates them to the indirect light ray-database accordingly.

3.3.1 Working Principle

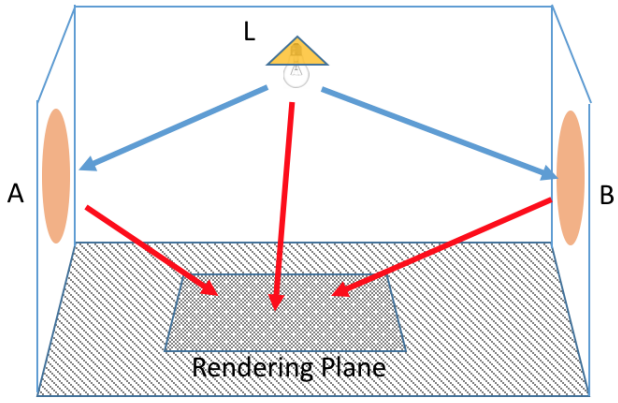


Figure 3.3: A schematic diagram of light distribution in a scene. Red arrows represent the light rays that ILF records. Blue arrows represent incoming light to the scene geometry from the light sources. ILF do not record these light rays.

ILF only contains information about incoming light from the environment to the capture space, not the light that the various parts of the scene receives. For example, the red arrows in the schematic diagram in Figure 3.3 are the light information that are known to the captured ILF in this scene. The blue arrows denotes the light flow towards the scene geometry which are not known to the ILF. The objective of DCP is to estimate the light information that the blue arrows carry, i.e. estimating the amount of light that the scene geometry receives from the light sources. This process is repeated each time the light source is changed synthetically in the ILF database.

DCP divides the scene into a number of discreet sections and estimates the total incoming radiance (called baseline radiance or L_b) on these sections for the

first static ILF capture with light sources unchanged. Suppose, on point X in the environment, the total estimated incoming radiance is $L_{b,x}$. The corresponding outgoing radiance from point X is $L_{o,x}$ which is already known and stored in the ILF ray database. After any change in the light sources, DCP calculates the changed incoming radiance (L_c) on every part of the geometry once more. At point X, the changed total incoming radiance now is $L_{c,x}$. For this work, DCP assumes diffused BRDF universally throughout the environment. Considering that the BRDF Bartell et al. [1981] remains the same during the change, the resultant outgoing radiance $L'_{o,x}$ can be calculated from the following relation.

$$\frac{L_{o,x}}{L_{b,x}} = \frac{L'_{o,x}}{L_{c,x}} \tag{3.1}$$

Chapter 5 describes the DCP technique in much greater detail including evaluations and validations of the technique compared to the real scenes. The next section provides an introduction to the Temporal ILF.

3.4 Temporal ILF

As already mentioned in Section 3.1, it is infeasible to capture Temporal ILF by physically capturing snapshots of static ILFs periodically. The proposed solution is to capture the static ILF once and track the changes in light simultaneously, eventually simulating these changes synthetically in the static ILF via DCP.

3.4.1 Working Principle

The general Temporal ILF working principle consists three stages. The first stage is to capture a low resolution geometry of the scene and subsequently capturing a static ILF in the scene. It is important to capture all the direct light sources that is contributing to the indirect lighting. The second stage is to track the light changes over time and represent them in an efficient format for the renderer. The third and final stage is to implement DCP technique on the captured static ILF and simulate the changes recorded in the stage 2. Chapter 6 describes all of these stages of the Temporal ILF in detail.

This chapter provided a brief overview of the entire workflow and the interconnection between the individual works that has been described in this thesis in order to develop and implement the Temporal ILF. The following chapters will describe these individual modules of work in much greater detail.

Chapter 4

Rendering with Incident Light Fields

"I don't think there is any truth. There are only points of view."

– Allen Ginsburg

Using images of real-world light sources has been an effective means to illuminate synthetic scenes in process of achieving photorealistic rendering. The IBL technique was discussed in Section 2.7 which captures a single light probe (a 360° HDR image of the environment) of the scene to render a synthetic scene illuminated by real-life light [Debevec, 2008]. However, in many situations light varies from one place to another. Focused lights such as spotlights and cast shadows produce *spatially varying light* in real-world scenes. The Light Field (LF) [Levoy, 2006; Levoy and Hanrahan, 1996] and Lumigraph [Gortler et al., 1996] techniques, discussed in Section 2.9 and Section 2.8 respectively, are capable of capturing the spatially varying appearance of a scene which enables these techniques to reconstruct the scene from arbitrary viewpoints. The Incident Light Fields (ILF) technique [Unger, 2009; Unger et al., 2003] has similarities with both the IBL and the LF, and is capable of capturing spatially varying incident illumination.

Section 4.1 provides a discussion of the prior techniques for rendering synthetic objects into real scenes as well as describes the general ILF working principle and the ILF pipeline in contrast to the previous works. Section 4.2 describes the various means of capturing ILF, their implementations and the limitations. Section 4.3 describes the representation and processing of the ILF data as well as the rendering techniques and describes an ILF representation method different than the ones described in literature. This new representation doesn't affect the rendering accuracy in any way but instead, suits the development of the Temporal ILF. Lastly, Section 4.4 discusses the limitations of ILF, possible ways to address them and provides a

summary to the entire chapter.

4.1 Background

The objective of the ILF technique is to provide means for photorealistic rendering of synthetic objects with real-world light so that they can be integrated into the real scene seamlessly. Section 4.1.1 provides a background to the ILF by discussing the prior works which deals with similar goals of realistic rendering of synthetic objects into real scene. Section 4.1.2 discusses the similarities and the contrasts between the ILF, LF and IBL. Section 4.1.3 describes the working principle of the ILF from the perspective of the plenoptic function and describes the general pipeline for the ILF.

4.1.1 Rendering Synthetic Objects

Efforts to integrate real-world images with synthetic objects deal with two primary types of issues namely the *illumination issues* and the *geometry issues*. The illumination issues deal with the computation of the the direct and global illumination of the synthetic object form the real-world environment. The geometry issues deal with the visibility and viewing parameter calculations which are important in Augmented Reality (AR) applications. Additionally, there are a number of other issues that need components of both illumination and geometry data. For example, one of the earliest efforts by Nakame et al. integrated computer generated 3D buildings into real-world images which also incorporated shadows of the synthetic buildings into the image [Nakamae et al., 1986] by utilising both illumination of the environment and the simple geometry of it.

Modern applications of AR is much more complex than simply putting a 3D model between the background and the foreground of a landscape. Complex indoor geometry, material properties and real-world illuminants significantly affect the illumination of a synthetic object. Fournier et al. further augmented from composited landscape images to develop a precursor to AR applications where synthetic objects are seamlessly integrated into a real indoor scene with complex geometry [Fournier et al., 1993]. This work involved obtaining the geometry and the position of the light sources in the scene (with a reasonable resolution), calculating the surface reflectances of objects inside generalised, imaginary bounding boxes from the available geometry and the illumination and then calculating the global illumination of the composited scene by re-rendering it with Radiosity algorithm [Cohen and Wallace, 2012] to produce a photorealistic image of a real scene composited with synthetic objects.

The main drawback of Fournier's method was its dependability on the manual input geometry for illumination. The user needed to provide the geometry of every

objects in the scene for the global illumination calculation. The Image Based Lighting (IBL) technique by Debevec et al partially solved this issue by capturing a HDR light probe as an environment map to illuminate the synthetic object without the need of the scene geometry[Debevec, 2002]. Section 2.7 in the previous chapter has described IBL in more detail; how IBL illuminates synthetic objects by dividing the scene into the local (near) scene and far-away (infinity) scene. However, compositing the synthetic object into the scene still requires *a priori* information about the local scene geometry from the user in IBL. Moreover, it does not take account of cases where there is a light source in the local scene either. This limits the effectiveness of IBL for applications where spatially variant lighting, cast shadows and caustics are essential for photorealism.

Sato et al solved this drawback by calculating the geometry of the scene automatically with an omnidirectional stereo image pair [Sato et al., 1999]. The process starts by constructing the geometry of the scene and then superimposing HDR images of the various parts (faces) of the scene geometry in place, effectively making a radiance map. This method is one of the most notable predecessor to the LF and the ILF.

4.1.2 The IBL, the LF and the ILF

The IBL technique [Sato et al., 1999; Debevec, 2008, 2002] is the most widely used technique to produce photorealistic renderings of synthetic objects to integrate them into a real scene. IBL is the easiest and least labour intensive to capture among the other two techniques. Rendering with IBL is also the fastest. In comparison, capturing the LF is expensive and with some methods, extremely labour intensive and slow. The same can be said with the ILF capturing techniques which is very labour intensive as well as requires expensive equipments. Although rendering with LF is not possible, ILF renders can be made real-time with proper representation and rendering technique.

Despite the ease of implementation, the major drawback of the IBL however, is their single point light capture. As described before, the radiance from a single light probe captured in high angular resolution is used as the light source. This allowed realistic natural light on one point but these approaches are unable to reproduce spatially varying light such as spotlight and cast shadows where incident radiance varies from one spatial point to another.

The ILF technique has been designed to solve this problem and is capable of rendering synthetic objects with complex light. It uses the LF approach to produce a dense distribution of light rays coming to every point in the rendering space from the environment as opposed to a single point light capture of IBL. Figure 4.1 provides an example comparison between rendering spatially varying light with IBL and ILF.

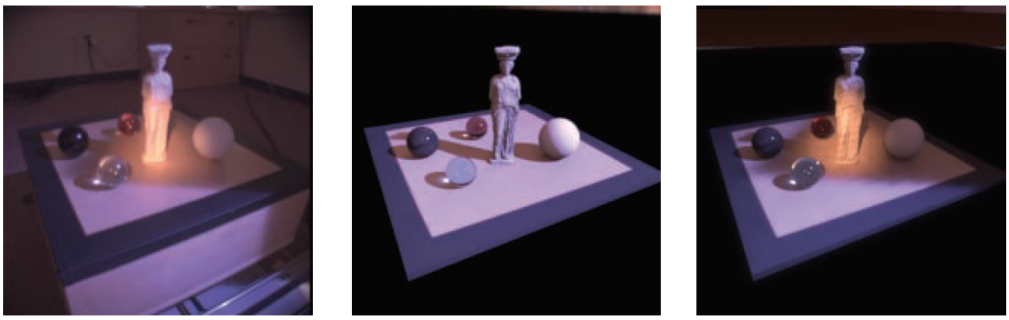


Figure 4.1: Identical scene rendered with IBL and ILF. (Image courtesy: Unger et al.)

In the photographed image, the scene has spatially varying light where the light at the center is different than the other parts on the table. While ILF can reproduce the lighting, IBL can't do that with only one light probe.

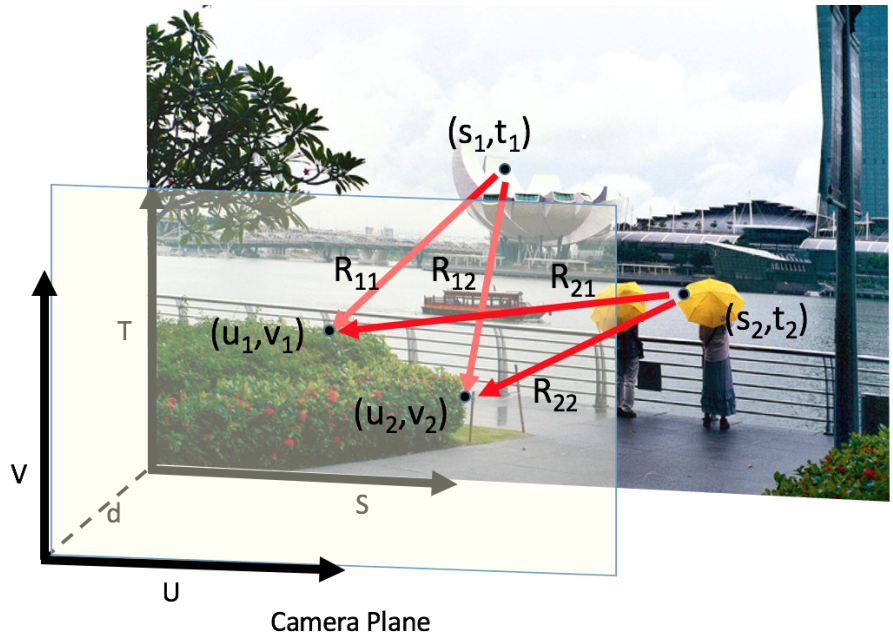
The operational similarity aside, the fundamental difference between LF and the ILF is in the objective of these two techniques. While ILF renders synthetic scenes with real-world illumination similar to the IBL, the goal of LF is to *reconstruct* a captured scene with additional functionalities.

Figure 4.2 further elucidates the design principle of ILF as well as the relations between ILF and LF. Figure 4.2a is a schematic diagram for LF. The Light Field of the scene in the ST plane is captured from the UV plane or the camera plane. Light rays depicted as red arrows can be seen coming from different points in the ST plane into different points in the UV plane. On the other hand Figure 4.2b shows the schematic representation of the ILF. The spherical environment has three light sources namely L_1, L_2, and L_3. There is an occluder present in the environment which prevents light rays from coming from the light source on the UV plane (shown as grey arrows), thus casting a shadow to produce spatially varying lighting on the UV plane. While the process of capturing the light rays are the same for both of the techniques, LF captures light coming from a scene while ILF captures the incident illumination coming from the environment.

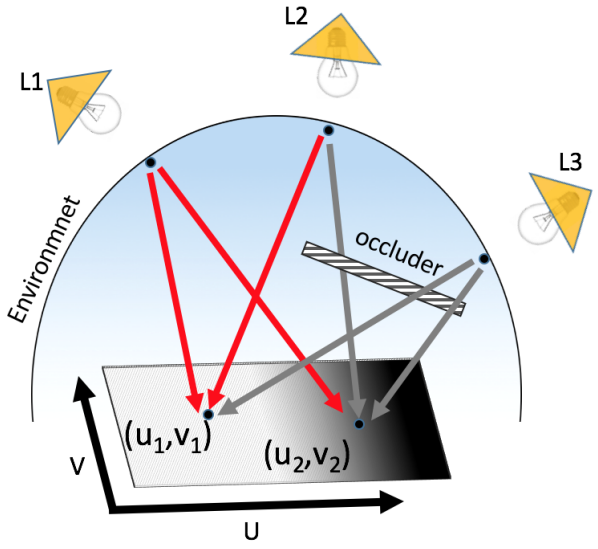
The next section will describe the working principle of the ILF from the point-of-view of the plenoptic function [Adelson and Bergen, 1991].

4.1.3 ILF Principle

Recalling the plenoptic function described in Section 2.6.1, light rays can be represented in a viewpoint independent manner with a 7 dimensional function $P = P(x, y, z, \theta, \phi, t, \lambda)$ where (x, y, z) is the origin position, (θ, ϕ) is the angular direc-



(a) A schematic representation of Light Field. The light rays (red arrows) come from the ST plane or Image plane (the scene) to various points in the UV plane, which is known as the camera plane.



(b) A schematic diagram of ILF. The environment has three light sources and an occluder. Red arrows are the un-occluded light rays and the grey arrows are the occluded rays. ILF captures the incident illumination on the UV plane.

Figure 4.2: Simple schematic diagrams of LF and ILF.

tion, t constitutes a moment in time and λ being the wavelengths of the light which is typically ignored in most cases. The IBL technique has a single point light capture which reduces the resolution of the plenoptic function for IBL to $P = P(\theta, \phi)$ ignoring the wavelength. As ILF captures a spatial distribution of light rays, the plenoptic function for ILF is given by $P = P(x, y, z, \theta, \phi)$.

Figure 4.2a presents a simple schematic diagram of the ILF principle. Points (U_1, V_1) and (U_2, V_2) has different amount of illumination coming from the environment. The arrows representing the light rays can be represented with the plenoptic function for ILF mentioned above. In order to capture such a spatial distribution of light rays, the ILF technique captures light probes from all possible points throughout the rendering surface (the U-V plane in Figure 4.2b).

There are a few different ways of representing and storing the captured light probes. The simplest way which was first implemented by Unger et al [Unger, 2009] stores the light probe images with their respective positions on the rendering plane. When a ray interacts with an object during rendering, the reflected ray is projected backwards and the appropriate light probe is found by the *Backprojection method* described in Section 4.3.2.

After this initial implementation, other versions of the ILF such as the "free-form ILF" [Unger et al., 2008] processes the light probe images and maps them in a bounding geometry. around the rendering space. This has been described in detail in Section 4.3. The work described in this thesis implemented a version of ILF which adheres to the "free-form ILF".

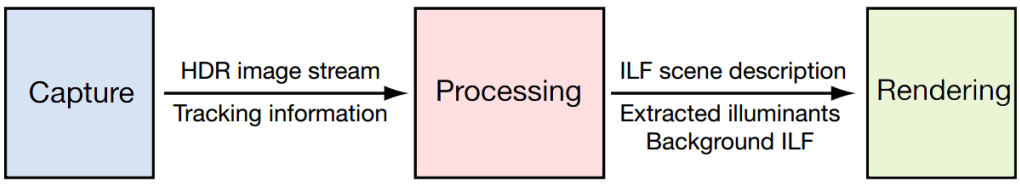


Figure 4.3: The general pipeline for ILF.

The general pipeline for ILF can be broadly categorised into three stages namely, capture, representation and rendering. Each of these stages can be further divided into specific methodologies and techniques used for each stages. Figure 4.3 describes the pipeline in a schematic diagram. The next chapters will describe the Capture, Processing and the Rendering with the ILF technique.

4.2 Capture

As mentioned in the earlier sections, there are a few different types of ILF in terms of its representation and capture methods. The basic principle for capturing the ILF however is the same for all of those approaches. Theoretically, an ILF consists of the incoming light rays coming from all directions to all possible points on the rendering space. The following sections will describe the few relevant methods and their limitations as well as the design decision to choose the best suited method among them for the present work.

4.2.1 HDR Light Probe Method

The most widely used method to capture the ray distribution in a ($360^\circ \times 180^\circ$) angular space is the light probe. Section 2.6.3 describes light probes in detail. It can be captured by taking a picture of a highly reflective mirrored sphere or a camera with a fisheye lens in it. The result, either way, is a spherical environment map.

As ILF captures the light sources directly, it is crucial to capture HDR environment maps. The LDR light probes do not have the required dynamic range to represent the very high energy light sources and the moderately lit or the shadow regions of the scene simultaneously. There are two primary ways to capture capture multiples of hundreds of HDR light probes: HDR video and HDR still images.

4.2.1.1 Light Probe Sequence

The earliest ILF capture methods used a sequence of light probes in one dimension (for example, only along X-axis) to obtain the ILF along the path which then was extrapolated to a 2D space [Unger et al., 2007]. Later the same method was upgraded to a 2D mirror sphere array to obtain the ILF. The grid of mirror spheres were placed on a surface and a high resolution HDR camera is used to take a picture of them thus, having a spatial distribution of light probes.

Although this is a fast and effective method of capturing ILF, the mirror sphere array is not at all portable. Moreover, the spatial resolution depends on the diameter of the each light probe. If it is too big, the spatial resolution suffers while too small means worse angular resolution.

4.2.1.2 HDR Video

As mentioned earlier, the ideal ILF capture would record a light probe on every possible point in the rendering space. In practice however, capturing light probes from every point in a given space is not feasible and thus the light probes are captured with a decent spatial resolution compromised between feasibility and effectiveness. The density of these captures, i.e. the number of light probes in a unit square area,

Figure 4.4: A mirrored sphere array. This can be used for ILF captures.

can be uniform as implemented in the light probe sequence [Unger, 2009; Unger et al., 2007]. It also can be "free-form" [Unger et al., 2008] as implemented in their more recent works.

Figure 4.5: HDR video camera attached to a light probe used by Unger et al. [Unger, 2009]

HDR video cameras with a decent frame rate are used exclusively to record ILF with the desired spatial resolution within a short time and effort. The camera is either used with the mirrored sphere or on its own with a fisheye lens. For both cases the camera is moved throughout the rendering space to record the light probes with the desired density. This eliminates the problem of insufficient spatial resolution in the light probe sequence methods.

The only problem with this is the availability of a good quality HDR camera with a decent frame rate so that moving the camera with a reasonable speed throughout the space to obtain the effective light probe density is feasible. Good quality HDR cameras are still very expensive and commercial high end cameras such as ARRI Alexa usually are not portable enough for such captures. The other option in this scenario is capturing still images.

4.2.1.3 Still HDR Images

In cases where HDR video cameras with the desired functionalities mentioned above are not accessible, the ILF is captured by capturing the light probes by manually taking still HDR light probes, move the camera manually around and throughout the entire rendering space.

The primary problem with this is the monumental effort required to complete a capture. A dense ILF capture done even on small places such as regular tabletops of $1m^2$ area can be upwards of 50,000 individual light probes. It typically take multiple hours just to capture one ILF with this method.

Due to unavailability of the hisgh-end HDR video cameras for this work, all of the real-world ILF captures in this work has been done with the still HDR light probe method. The extremely time consuming process was a hindrance for achieving the spatial density typically used by the ILF literature [Banterle et al., 2017]. It was also quite difficult to determine the exact spatial positions of each individual light probes. Section 4.2.3.3 describes an approach to mitigate the position tracking problem using fiducial markers.

4.2.2 Planar Mirror Method

Apart from the light probe approaches, there was an alternative approach proposed by Ihrke et al which uses coded planar mirrors [Ihrke et al., 2008] instead of using the light probes. The edges of the mirror is binary coded to determine the exact angle of the mirror. During capture, the mirror is moved around and tilted to capture the total angular distribution of the upper hemisphere of the rendering space.

There a re a few limitations with this approach. Firstly, equipment wise, a large mirror without aberrations is quite expensive. It is not exactly portable either. Tilting the mirror either will need someone to manually do the job or a specialised gantry system to do this automatically. *Secondly* This again uses HDR video cameras for fast capture while with a fisheye lens on such a camera, there'll be no need for the mirrors as it is a matter of moving the camera around as done in [Unger et al., 2008].

4.2.3 Position Tracking with Moving Camera

The moving camera methods capture light probe images throughout the rendering space by either moving the camera or the light probe and camera pair together. In the case of HDR video captures, each of the frames containing light probes must have their global positions. Precision is important with the position detection specially while capturing complex lighting and lights with hard edges such as a spotlight.

ILF captures are usually done with variable capture density throughout the

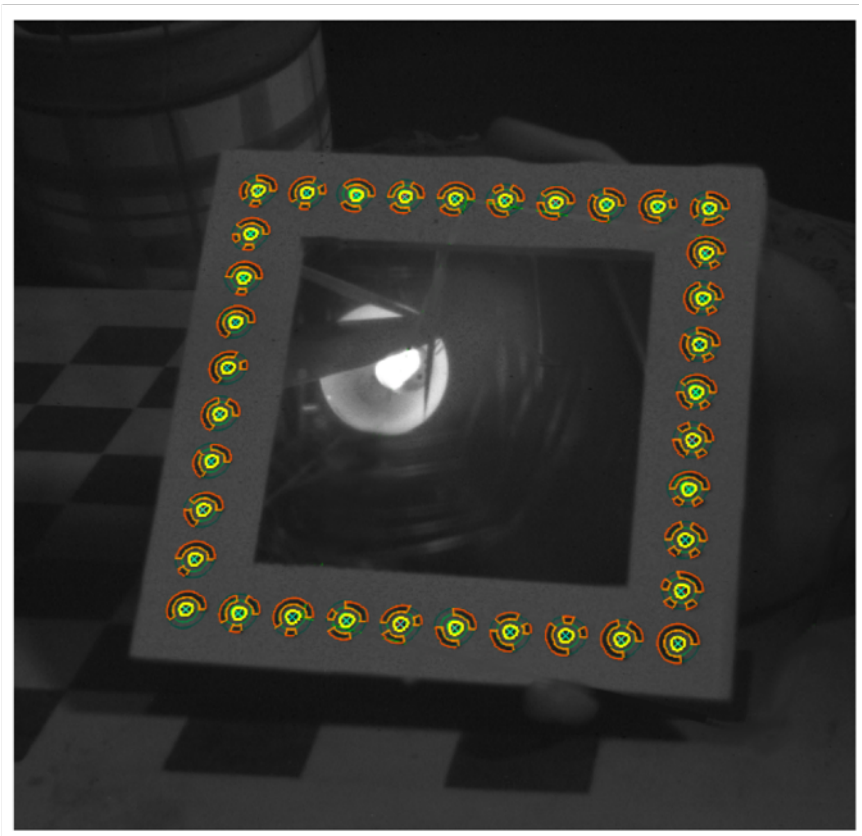


Figure 4.6: The planar mirror method of ILF capture. (Picture courtesy: Ihrke et al.) [Ihrke et al., 2008]

capture area. The parts of the scene with intricate lighting which changes rapidly within a small area are captured with the highest possible light probe density.

There are a few ways to determine the positions. The position detection in small scale captures need to be more precise as they capture intricate lighting with high spatial density over a small area or volume. Large area captures will need a slightly different approach and need not be very precise in their position detection as they do not capture intricate light and shadow details, but the general direct and indirect light. Below are the relevant position tracking systems.

4.2.3.1 Motorised Gantry

Unger et al developed a motorised gantry [Unger et al., 2003] which is capable of moving the camera within its range of operations. This gantry was fitted with an "opto-mechanical" device which track the position of the gantry against elapsed time. According to the work, it supposedly showed a very good detection accuracy, enough to record ILF with high spatial resolution.

One major limitation of this would be the portability of the system. Moreover, this approach would also be tedious while capturing spaces which are larger than

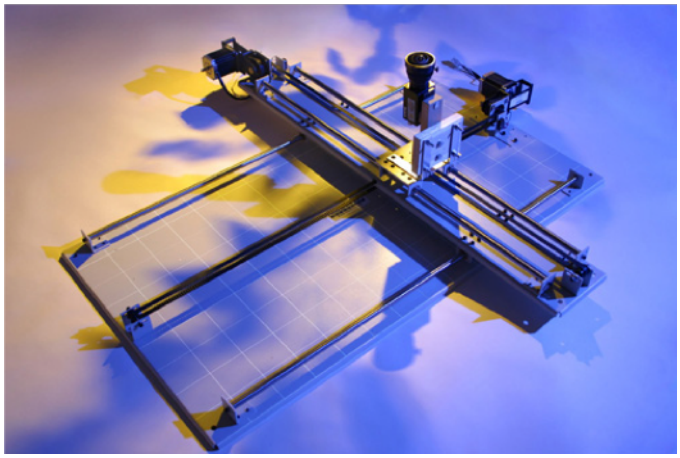


Figure 4.7: Motorised gantry used by Unger et al. (Picture courtesy: Unger et al.) [Unger, 2009]

the operating range of this. Lastly, there were not much information about the "opto-mechanical" position tracker used with this system nor this was commercially available. The ILF captures presented in this thesis thus chose to implement other means of capturing.

4.2.3.2 Opto-digital Tracking

As there were no clear knowledge of the tracking method used in the literature apart from the mere mention of them, this work developed a "opto-digital" tracking system which uses a standard 500DPI optical mice sensor for position tracking. The sensor is housed in a metal container specifically designed for this purpose with added means of attaching a camera to it in order to move it around freely.

A custom software has been developed in order to process the raw data form the mice because the operating system always add a mixture of acceleration functions and scaling functions on the raw data which makes it unusable for tracking.

The major limitation of this approach was its inaccuracy while moving the camera in arbitrary directions. On testing it was found that the mouse data is only reliable when it is either moved sideways or vertically relative to the optical sensor. This approach would have worked well with a 2dimensional gantry operating on a surface similar to the one used in Unger et al's work described above [Unger et al., 2003; Unger, 2009]. The second major limitation of this approach was it always needed a surface to track the position, as it effectively was a computer mice sensor. Captures in 3D space would have been impossible with this method.

4.2.3.3 Marker Detection Based Tracking

Due to the unavailability of a 3D motorised gantry and the major limitations of the opto-digital tracking developed, this work implemented a marker detection based tracking where a marker image is placed in the environment which can be "seen" by the light probe images from the capturing area.

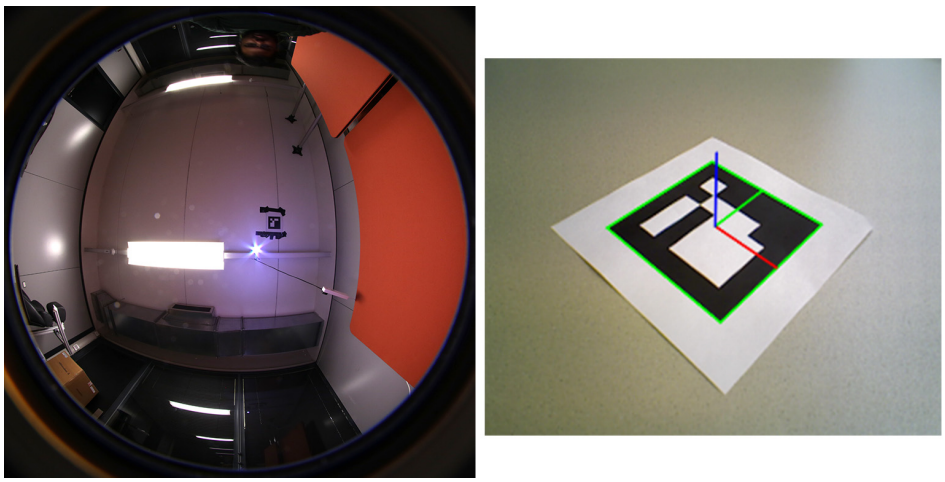


Figure 4.8: Marker detection based tracking. The left image is a light probe from one of the real-world testing. Notice the marker image in the center of the ceiling. Right side is a marker detected by a in-house detection software written in C++ and OpenCV.

For the marker, the commonly used fiducial marker system "ArUco" [Garrido-Jurado et al., 2014] has been used. These types of marker images can simply be printed in paper and easily attached to any surfaces. ArUco system has one of the best detection accuracy [La Delfa et al., 2015] and are widely used in the computer vision applications.

During the capture, the marker image needs to be placed in a position inside the environment where it can be fully captured by the light probe images. A custom software has been developed to detect the marker in the light probes using the existing ArUco detection library (OpenCV implementation in C++) and subsequently to calculate the position of the light probes from it.

4.3 Processing and Rendering

This section provides a description of the various representational techniques and the methods of rendering with the ILF.

4.3.1 Processing and Representation

As mentioned earlier, the captured light probe images are processed and represented in a way which is easier to store, edit and retrieve during the rendering. The first step is to create a ray-database from the light probe images. There are a few different ways to store this ray database next. The representation relevant to this work is the ILF slab and the Source Light Field (SLF). The next sections will describe them in more detail.

4.3.1.1 Ray-Database

Recalling the plenoptic function, the distribution of the light rays are represented as $P = P(x, y, z, \theta, \phi)$ or in short, $P = P(x, \omega)$ where ω is the direction and x is the origin position. Each pixel in the captured light probe images represents a light ray with direction ω. The capture positions serve as the origin position x for the particular ray. This way, every pixel in every captured images can be represented as an individual light ray. The whole spatial and angular distribution of these light rays are stored in a ray-database which is easier to retrieve during rendering.

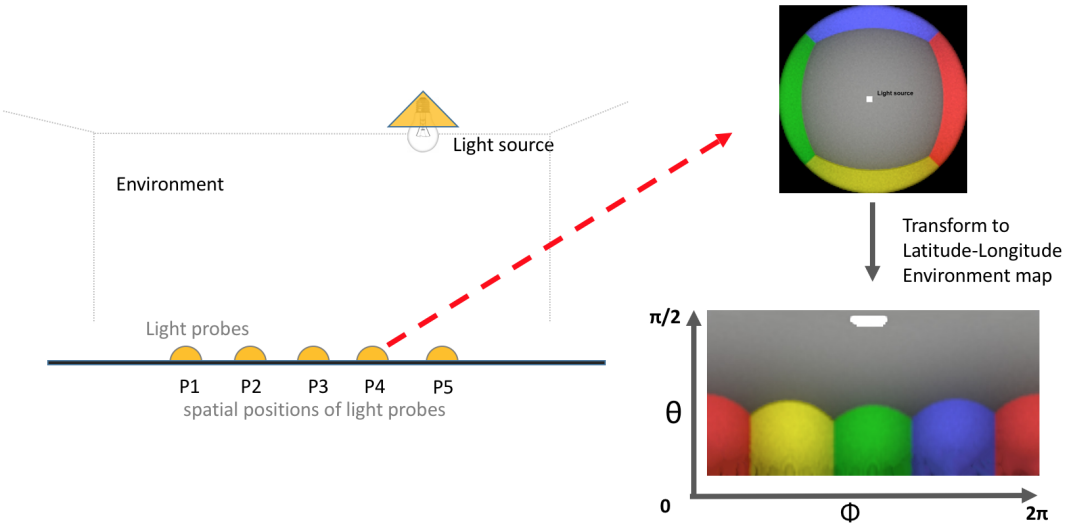


Figure 4.9: Light probes to a ray-database. The environment has the light probes at position $P_1 - P_5$. A sample light probe from the position P_4 is shown which is converted to a Lat-Long environment map. Each pixel on this lat-long map is a light ray with direction (θ, ϕ) and the origin position P_4.

There are a number of ways a ray-database can be constructed based on the intended representation and application. The original literature describes methods such as the *Ray-binning* where the rays are grouped and stored by their angles so that every ray in an angular bin has the same angles but different origin positions.

Another approach makes use of volume based representation of the light rays in *Energy voxels*. As these approaches are out of scope for this work, details of both can be found in the original ILF literature [Unger, 2009].

This work represents the light rays in a ray-database where the light rays are grouped spatially. The spatial dimension in question can either be the captured spatial dimension in case of the simplest implementation of ILF described in the literature or a projected dimension on an imaginary plane. The later is achieved by a technique called the "ILF Re-projection" which is described next.

4.3.1.2 ILF Re-projection

The captured light probes often are irregularly spaced which results in a large processing time during rendering to find the correct light ray with the correct origin position from billions of light rays. For this reason, the ILF ray distribution is projected to an imaginary bounding-box enclosing the rendering space. Each side of these boxes are called the **ILF plane** [Unger, 2009]. The dimensions of the bounding box can be arbitrary as long as it encloses the capture area. This method has been described as the ILF re-projection in the literature [Unger, 2009]. Figure 4.10 describes the re-projection of light rays to the ILF planes $(P_1, P_2, P_3, P_4, P_5,)$ surrounding the capture area A.

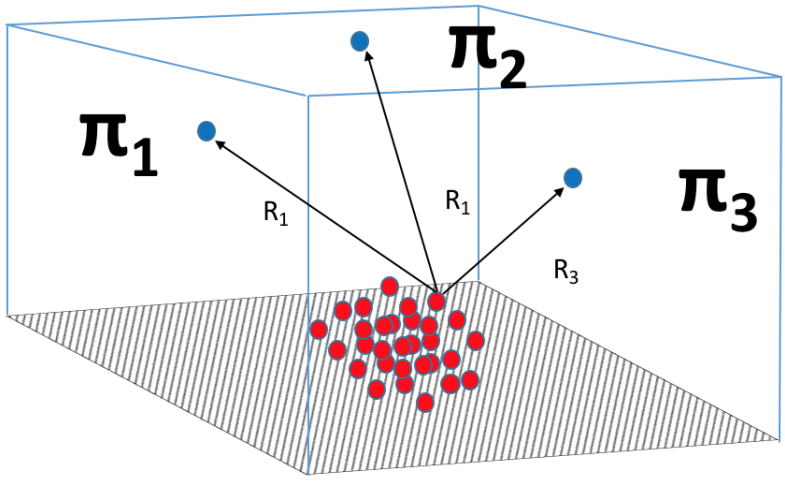


Figure 4.10: The ILF re-projection. The red dots are the light probe samples in the shaded capture area. The bounding-box around the capture area are made of "ILF planes" named Π_1 for the back side, Π_2 for the ceiling, and Π_3 for the left side in this diagram. The rays from the light probes are reprojected on these planes. The new projected rays will have the new origin positions (blue dots) on the ILF planes.

Apart from the ILF planes, Unger et al. also described a method to extract the illuminants from the ILF ray-database by a semi-manual process. This process uses thresholding to determine high energy rays and group them together as a light source while the operator (the user) manually decides the actual position and size of the light source from a visualisation. These high energy rays are stored separately in a similar spatial data structure as ILF planes and are called **Source Light Fields (SLF)**.

Originally the ILF planes do not need to adhere to any specific dimension as long as the bounding-box is convex to the capture space. Only The SLFs needed to be in the correct place. However, for the specific purposes of developing the Temporal ILF (described in Chapter 6), the ILF planes are made with the dimensions from the boundary of the capture scene. For example, even if the ILF capture space is a small part of the entire scene, the ILF planes are made according to the boundary of the scene and not any arbitrary convex bounding box. This does not affect the rendering quality as the it is not dependent on the size of the position of the ILF planes. It however facilitates mapping the spatial ray-data database easily to the scene geometry. This is a crucial requirement of the works that has been presented in this thesis. Moreover, the SLFs are implemented in this implementation but the illuminant extraction technique is different from the literature which is further described in the next chapter as a part of the DCP technique. More clarification on this particular design decision has been discussed in both Chapter 5 and Chapter 6.

4.3.2 Rendering

After the processing and representation of the ILF data, this section describes the pertinent ways to render with an ILF. Chapter 2 has described the rendering equation and various rendering techniques in detail. A simplified rendering equation is given by:

$$L_o(x, \omega_o) = L_e(x, \omega_o) + \int_\Omega f_r(x, \omega_o \leftarrow \omega_i) L_i(x \leftarrow \omega_i)(N_A \cdot -\omega_i) d\omega_i \tag{4.1}$$

Here $L_o(x, \omega_o)$ is the outgoing radiance from point x towards direction ω_o. $f_r(x, \omega_o \leftarrow \omega_i)$ is the SBRDF, $L_i(x \leftarrow \omega_i)$ is the incoming light to point x in the direction ω_i. N_A is the surface normal of the area surrounding point x. As a ray is hit on a point x in the scene, the reflected direction ω_i is sampled, and subsequently the ILF ray-database is queried for the incoming light $L_i(x \leftarrow \omega_i)$ given (x, ω_i).

Originally there are three different rendering methods for 1D, 2D and 3D free-form ILF. For the 1D and 2D ILF, the basic rendering process makes use of a *back-projection* technique. For a reflected ray R from a hit point X in the synthetic scene, with the outward direction ω towards the environment, the ray R is extended

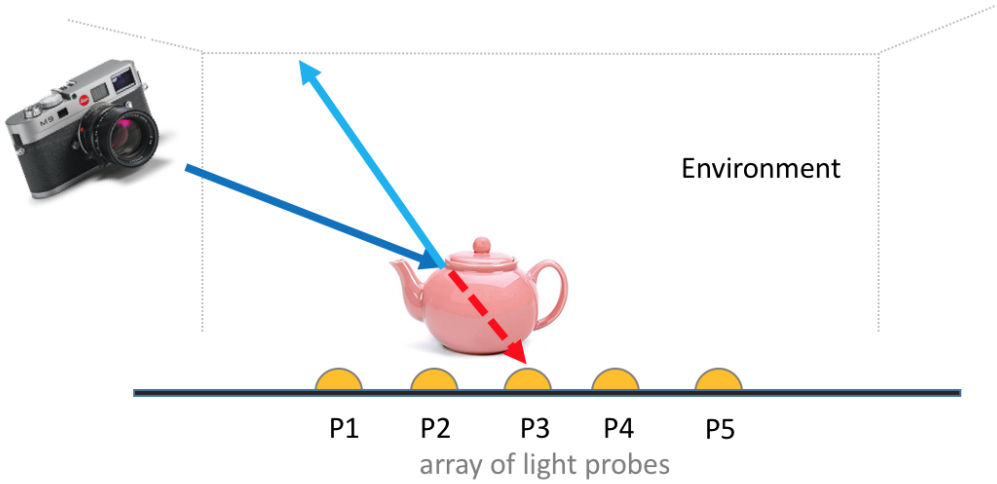


Figure 4.11: The ILF back-projection. During rendering, when a ray from the camera (dark blue) is hit on an object, the reflected ray (light blue) is extended backwards (red) to find the proper position of the light probe for that ray.

in the opposite direction $(-\omega)$ of its original direction ω so that the corresponding hitpoint x' could be found and subsequently the corresponding light probe could be accessed.

This work however implements the rendering methods used for 3D free-form ILF which takes the advantage of the ILF planes and the SLFs. This method uses the hit point and the ray direction (x, ω_i) to find the intersection point x' in the ILF planes or the SLFs. Figure 4.12 shows a schematic diagram of the rendering with the ILF planes and SLFs. This section described the overview of the rendering methods without describing the detailed descriptions which are out-of-scope of this work.

4.4 Discussion

This section described the ILF technique in detail. As mentioned, the ILF implementation developed for this work is representationally slightly different from the original implementation of Unger et al to better suit the development of the Temporal ILF described in Chapter 6. Figure 4.13 shows a sample image of a synthetic scene rendered with the implemented ILF. The design decisions which were taken differently from the original ILF literature did not affect the quality of the images in any way.

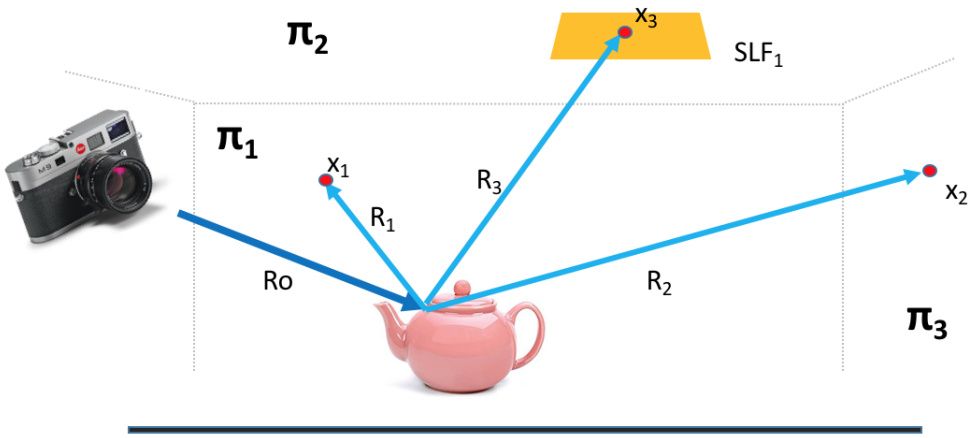


Figure 4.12: Rendering with ILF planes and SLF. The reflected rays R_1 and R_2 from the object hit the points x_1 and x_2 on two ILF planes Π_1 and Π_3. The ray x_3 hits one of the SLFs and the hitpoint is x_3.

4.4.1 Limitations

Firstly, one of the limitations of the ILF technique is the complexity of its capture and the implementation. This chapter already described the challenges of acquiring high precision instruments like HDR video cameras and custom capture rigs as 3D mechanical gantries in Section 4.2. Not only the instruments typically used are prohibitively expensive, but also they are not portable enough to be used easily for a general user. This makes the technique all but inaccessible to general consumers.

Secondly, another major limitation of the ILF technique is its lack of scalability. ILF has been designed for capturing smaller indoor spaces with intricate lighting. Although Unger et al's later implementations of ILF captured and produced promising photorealistic renderings of IKEA furnitures [Unger, 2009], the entire process was prohibitively complex to be used in mainstream applications. Moreover, even larger spaces in the outdoors such as racetracks will be completely unfeasible with the present technique. The huge ray-database overhead for such large ILF captures would be very hard to manage without compressing them significantly.

Thirdly, ILF can edit the colours and the intensity of the direct light synthetically by editing the LSFs. However the changes made in the SLF do not propagate to the indirect lighting stored in the ILF. Moreover, it is not possible to move the SLFs around the scene. The next chapter will describe a novel technique DCP which addresses this limitation of ILF and provides the ability to change SLFs with indirect light fidelity.

Finally, the primary limitation of the ILF that this work addresses is its in-

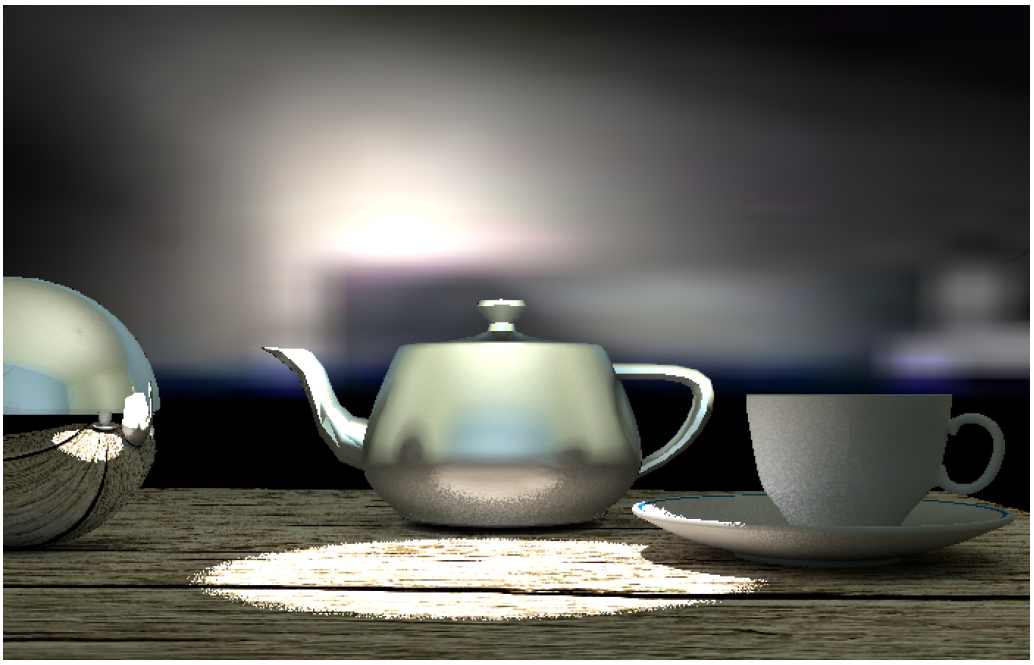
ability to capture and render from spatially varying light which changes dynamically. As mentioned in the Chapter 1 and 3, the ILF capture, even with the fastest possible methods, are incapable of recording dynamically changing complex lighting. This thesis describes a novel technique Temporal ILF in chapter 6 which takes inspiration from the ILF technique and provides means to capture and render from spatio-temporally varying light.

4.4.2 Summary

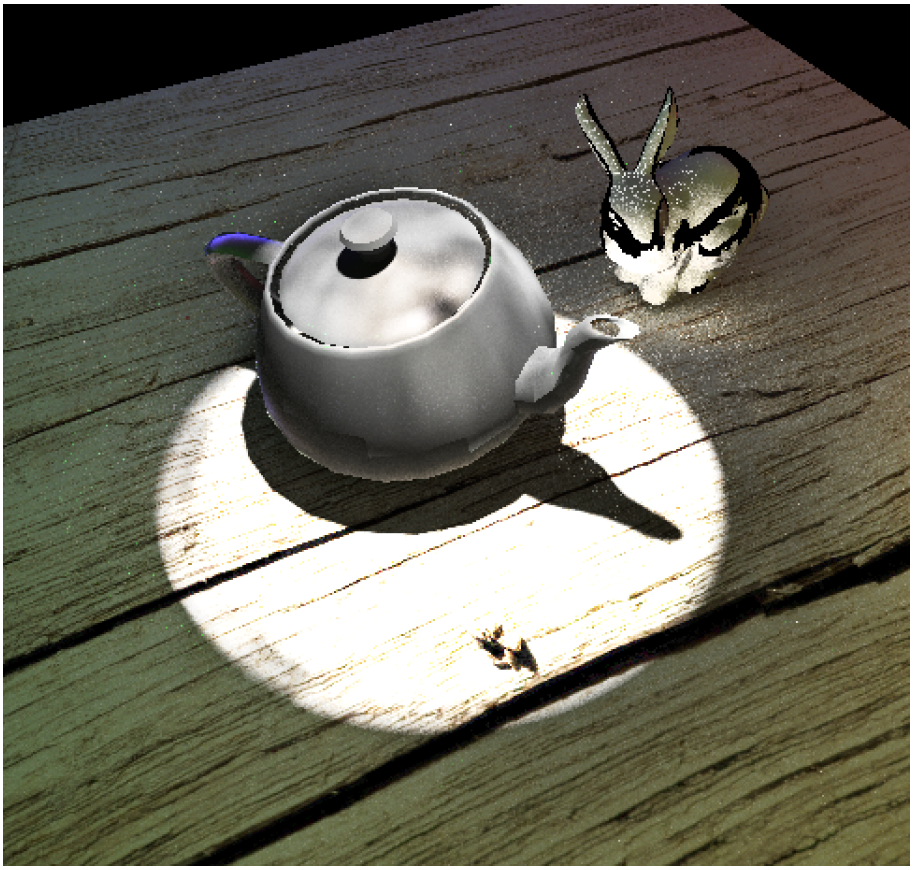
Following is the summary of this chapter:

- ILF is a technique which can capture and render from complex, spatially varying light.
- The ILF captures multiple light probes to produce a spatial and angular ray distribution which is reprojected into a set of bounding planes called the *ILF Planes*.
- ILF ray-database can be used to extract high energy light rays coming from the direct light sources. The extracted direct-light rays are mapped to a special type of ILF plane called the Source Light Fields (SLF).
- The light intensity and colours in the SLFs can be edited by the user.
- There are a few different implementations of ILF. This work implemented a version inspired by the free-form ILF where the ILF planes and the SLFs adhere to the scene geometry.

The next section will describe the Dynamic Change Propagation technique.



(a) ILF render from scene 1 with a spotlight. The parts the environment can be seen reflected in the synthetic mirrored ball.



(b) Render of scene 2 with another spotlight. The walls of the environment are red and green which can be seen in the colour bleeding on the texture.

Figure 4.13: Rendering synthetic objects with a captured ILF

Chapter 5

Dynamic Change Propagation

"This is the way I travel. Earth, Sun Stars accompany me."
– **Falguni Ray, *Personal Neon***

The ILF technique described in the previous section has very limited editing capabilities. It cannot change light source positions and the radiometric changes it can make to the light sources synthetically, is only local to the light sources and not the entire environment including the indirect light.

As described in Chapter 4, an ILF ray-database can be divided into high energy light source regions (SLF) and low energy background regions (indirect light reflections). This Chapter describes the Dynamic Change Propagation (DCP) technique which can synthetically change the light sources in an ILF with indirect light fidelity. DCP propagates the effects of any arbitrary change made in the SLFs by editing the rest of the ILF database containing the indirect light according to the change in the direct light.

Section 5.1 highlights the basic working principle of DCP and the previous works that relates to the DCP technique. Section 5.2 describes the various modules of DCP in detail. Section 5.3 describes the results of the effectiveness and accuracy of the DCP technique by comparing the rendered images to the ground truth for two virtual environments. Finally, Section 5.3.3 discusses the results, the limitations and the possible future works to mitigate the current limitations.

5.1 Background

This section will describe the basic working principle, the application areas and the works prior to DCP that partly relates to its methodology briefly.

5.1.1 Working Principle

5.1.1.1 Problem Statement

ILF captures the ambient light information of both direct and indirect light coming from the surrounding. For a given point x in the scene geometry, ILF stores the outgoing radiance (L_o) from x but has no information on how the light from that point is being emitted or reflected. It is possible to determine whether point x is a light source, however, in case x is not a light source and just reflecting indirect light, ILF does not have any information about the amount of incoming light that point x receives.

Assuming point x in Figure 5.1 is a part of the scene geometry reflecting light, ILF only captures the light coming form x, drawn in the diagram as red arrow. It does not capture the light coming to x along the blue arrows. An estimation of the incoming radiance L_i to the point x is crucial since any change in the light sources will change the L_i and subsequently, the reflected indirect light L_o from point x should also change accordingly. Once again, Figure 5.1 further elucidates the scenario. The point x in the scene geometry in the diagram has the incoming radiance L_i and the reflected radiance L_o (shown with just one blue arrow for each position for simplicity). If the light source changes position to the right, the amount of incoming radiance will change from L_i to L'_i. Correspondingly, the radiance reflected from X should change form L_o to L'_o.

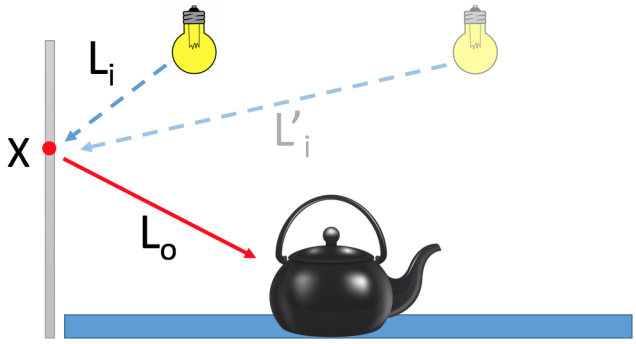


Figure 5.1: The light reflected back (L_o) from a point x in the captured scene should change according to the changes made to the light source as the incoming radiance L_i changes to L'_i with changed light position.

5.1.1.2 Proposed Solution

As ILF only stores the outgoing radiance L_o from any given point (in this case, point x), in order to calculate the updated radiance L'_o from x after a change in the light source, it is necessary to estimate both the original incoming radiance L_i and the changed incoming radiance L'_i.

For simplicity, Figure 5.1 only shows the direct incoming radiance to point x. In actuality, the L_i is the incoming radiance from multiple light sources i.e. direct light, as well as reflected light from the rest of the environment. The DCP technique described in this chapter estimates the total incoming radiance L_i in different parts of the captured scene and propagates any changes made to the light sources throughout the ILF database dynamically by re-calculating the L_i' every time such a change occurs and adjusting the outgoing radiance L_o accordingly.

5.1.2 Applications

The primary application of the DCP technique is Temporal ILF. However, It can also be used in object relighting in images and videos in a possible future work as well.

5.1.2.1 Temporal Incident Light Field

The primary objective of this work has been to produce a Temporal Incident Light Field (Temporal ILF). The limitations of static ILF has been discussed in detail in Section 4.4 and has been concluded that it is infeasible to capture ILF with traditional means in a scene over a period of time where the light source changes dynamically. As ILF can only record spatial changes, in order to capture ILF over temporal changes, it is imperative to devise a technique that calculates the effect of changes in the light sources over time. In such a scenario, the effect of the changes must be simulated by a dynamically calculated propagation technique.

5.1.2.2 Object Re-lighting in Images and Videos

A significant body of research has been conducted on relighting real-world images after they have been photographed. There are various approaches which acquire a loose 3D model of the photographed object to predict the position of the light source and changing the position or the radiometric properties of the light. Dutré et al. has extended the Lumigraph [Gortler et al., 1996] concept to manipulate the incident light on an object in Image based Lighting Design [Anrys and Dutré, 2004] approach. Another approach taken by Dutré relights objects with 4D incident light field data [Masselus et al., 2003] which is closer to the end goal of the DCP. However, the term incident light field used by that particular work is different than Unger et al's Incident Light Field [Unger, 2009] on which the current work is based.

Furthermore, there are other approaches that attempt to achieve post production relighting of photographs such as Bayesian Relighting [Fuchs et al., 2005] and several face relighting approaches [Wang et al., 2009], [Qing et al., 2005]. All of

these however have severe limitations in terms of taking into account the indirect, ambient light. The DCP technique can be very effectively used for post production image relighting of photographs, provided an incident light field of the environment has been captured. This will be further discussed in the Section 7.4 describing the ongoing and future works in the last Chapter of this thesis.

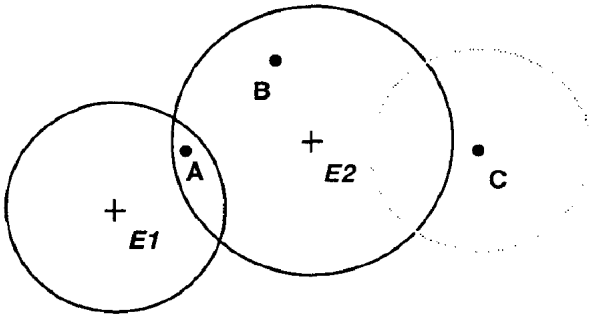
Similar to the still images, post-production relighting of video data in a motion picture scenario is an open challenge in the CGI application industry. The Temporal ILF as an extension to the DCP technique will be an ideal approach to achieve this. Section 7.4 explores this as a possible future work later.

5.1.3 Related Work

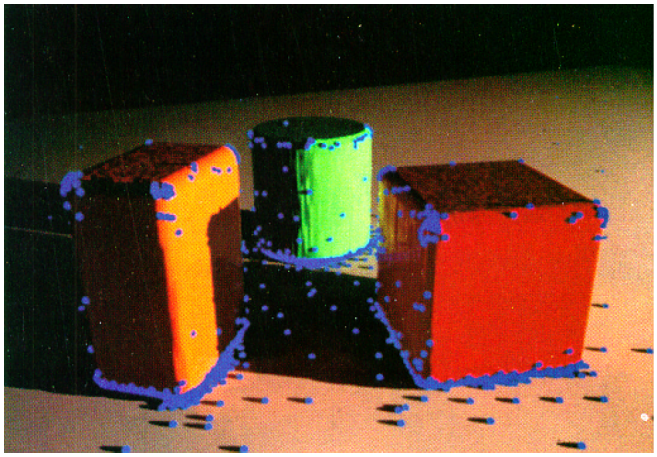
Although there are no other approaches for real-time recalculation of an ILF according to light source change, some previous research has been conducted which provides a basis for the DCP technique. One of the seminal methods among these is the indirect illuminance caching as proposed by Ward [Ward et al., 1988], who introduced an efficient way of ray tracing by storing the pre calculated illuminance on a few points in the scene to accelerate the calculation of diffused indirect lighting. Originally devised for a different purpose (out of the scope of this thesis), Ward's technique of storing the illuminance values at different points in the scene and averaging them to compute diffused indirect lighting for the points in the vicinity of these stored values is the precursor to the radiance caching techniques that has been popularised much later.

In Figure 5.2b, the cached points are shown with blue dots. The algorithm is described herein: After a ray hits the scene, the hit-point is checked as to whether it falls in the range of some previously cached points. If there are any cached points in the vicinity of the hit-point, then the illuminance of the hit-point is calculated by averaging from the nearby cached points. If the new point falls outside of the range of any such existing cached points then its diffused illuminance is calculated using the ray tracing based global illumination and stored (cached) into that hit-point as a cached-point. Figure 5.2a shows this schematically. E1 and E2 are cached points. If the new point appears at the position A, the illuminance is calculated from E1 and E2. However, for point B, the illuminance is the same as E2 since B's proximity to E2 is greater than that of E1. Furthermore, for point C, the diffused illuminance has to be calculated since it lies beyond the range of both E1 and E2, respectively. After the calculation point C will be added to the cached-point list.

Apart from the diffused illuminance caching, the other relevant method in the context of DCP technique described here is the Instant Radiosity (IR) technique proposed by Keller et al [Keller, 1997]. In this technique, the photons are traced from the light source to the various parts of the scene and the path vertices are



(a) Point E1, E2 has the diffused illuminance stored. Point A will calculate its diffused illuminance from E1, E2. B will calculate from E2. Point C will have to calculate it on its own and thus will store the calculated diffused illuminance at C.

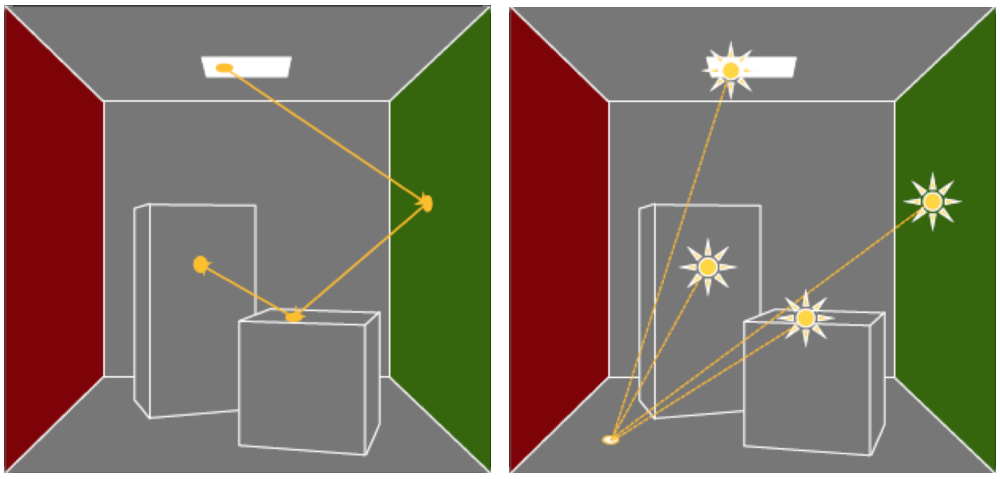


(b) The cached points (in blue) in the rendered scene.

Figure 5.2: The diffused illuminance caching. Figure from [Ward et al., 1988].

stored in an easy-to-access data structure as *Virtual Point Light* (VPL) sources. The scene is rendered for each of these VPLs and the shadow-rays are computed by treating each of them as light sources. Finally, these multiple rendered images are accumulated in the accumulation buffer to obtain the final rendered image. There are a few modern variations of the IR such as bidirectional IR or metropolis IR which applies better techniques to concentrate the VPLs more intelligently than a random sequence in order to achieve better results with less number of VPLs. The basic technique however remains mostly the same. a more detailed discussion of IR could be found in Chapter 2 of this thesis.

Parts of both the IR and Illuminance caching approaches are ideologically in the vicinity of the DCP technique as its central idea is to estimate the incoming illuminance in the various parts of the surrounding geometry and make necessary changes to the background ILF if the light source changes position, intensity or



(a) The VPL creation by tracing a ray from the light source.

(b) Treating each hit points as VPLs, raytrace from every point in the scene towards all of the VPLs and accumulate the illuminance.

Figure 5.3: The Instant Radiosity technique. Schematic diagram of a standard cornellbox scene.

colour. However, the context and the application of DCP is very different from both of these methods. The next Section 5.2 describes the DCP technique in detail.

5.2 DCP Technique

The DCP technique has four parts; (a) the light source and scene geometry detection, (b) distribution of update points (UPoints) around scene geometry, (c) calculation of estimated incoming radiance, and (d) updating the ILF database. Figure 5.4 provides a simplified flowchart for the DCP.

As the first step, the scene geometry of the environment along with the position, orientation and size of the direct light sources in the environment is detected. The scene geometry is then discretised into a number of sampling points named "UPoints" which are distributed along the scene geometry. UPoint density can vary in different parts of the scene. Subsequently, DCP calculates the estimated incoming radiance on each such UPoints. The fourth and final step in DCP is to use this data to update the main ILF database, i.e. propagate the changes.

In order to propagate the changes made in the light source to the indirect light stored in the ILF, the radiance estimation is done two times. First, a *baseline radiance* L_b is calculated with the original position at the start of the procedure. After every change made in the light source, the *changed radiance* L_c is calculated. The resultant ILF is adjusted accordingly. Algorithm 1 explains this procedure from a top level view of the DCP technique from the generation of UPoints. It is

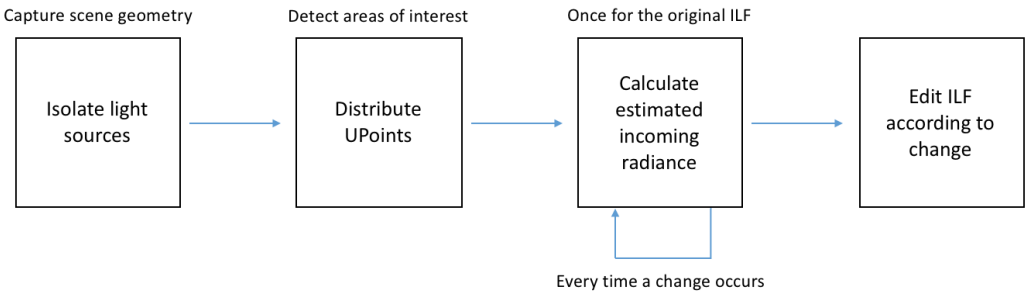


Figure 5.4: The DCP technique pipeline.

noteworthy that the DCP calculates the estimated baseline radiance L_b for the first time by ignoring the occlusion caused by the synthetic objects being rendered as they were not the part of the real-world scene. For all the subsequent times (for L_c) it takes account the occlusion caused by the synthetic objects being rendered for a more realistic result. Section 5.2.3.2 later explains this in more detail.

Algorithm 1 DCP Algorithm (Top Level)

```
1:  procedure DynamicUpdate(ILFhandle, Scene)
2:      generateUPoints();
3:      ignoreOcclusion ← TRUE
4:      L_b ← estimateRadiance(ignoreOcclusion);
5:      updateILFDatabase(L_b);
6:      while SLF change do
7:          updateSLFdatabase();
8:          ignoreOcclusion ← FALSE
9:          L_c ← estimateRadiance(ignoreOcclusion);
10:         updateILFDatabase(L_c);
11:     end while
12: end procedure
```

The next sections describe the four parts introduced above in greater detail.

5.2.1 Isolating Light Sources from Scene Geometry

As established by Unger [Unger, 2009] and described earlier in Chapter 4, the ILF need not be aware of the scene geometry for the purpose of rendering a synthetic object. It also does not particularly need to isolate light sources in the ray-database. However, doing so increases efficiency and decreases its render time significantly. In most real-world implementations of the ILF, Unger et al. included the scene geometry for better reconstruction of the light sources and is useful for conducting

any manipulation, local to a part of the scene [Unger et al., 2003, 2008].

In this work, DCP calculates the impact of the changes in direct lighting into the indirect lighting in a captured ILF of an environment. It automatically updates the luminance data in different parts of the environment. Thus correlating the ILF data with appropriate parts of the scene geometry is absolutely crucial for this work. Sections 5.2.1.1 and 5.2.1.2 next will describe how the scene geometry is detected and the light sources are isolated in it.

5.2.1.1 Scene Geometry detection

The acquisition of accurate scene geometry can be significantly time consuming manual work. Minor depth irregularities along the boundaries (for example, ridges, troughs, pillars, racks, etc along the wall in a room) can be captured with high resolution geometry data from expensive laser scanners and related equipments. However For most real-world situations, higher resolution geometry does not significantly improve the results than a low resolution estimation of the geometric boundaries of the environment. This hypothesis is supported by the evaluations in Section 5.3 where it compares the error rates of DCP with a high and a low resolution geometry and concludes that high accuracy in scene geometry acquisition is redundant in all but the extreme cases. This work thus uses a low resolution geometry data acquired with easy to implement and inexpensive, conventional methods.

There are various ways of acquiring a scene geometry depending on the size and complexity of the scene, the required precision and ease of implementation. As mentioned, large, intricate scenes can be captured in great detail with dedicated laser scanners [far]. However, this work uses similar techniques used by Unger et al [Unger et al., 2008] where well established Structure from Motion (SfM) based methods were used to acquire the geometry by capturing a few planer images of the scene around and processing them with available generic 3D reconstruction algorithms to get a point cloud and subsequently, the geometry of the scene. Figure 5.5 provides a schematic diagram for the standard implementation flowchart of such a geometry reconstruction method.

The first step of this process is capturing a pair of images with known distance between the two. The next step is to detect the edges, such as corners or some object along the wall, in the image pair by using existing robust feature detectors such as the Harris corner and edge detector [Harris and Stephens, 1988]. Subsequently, the detected features in both images are matched with each other by a standard block matching algorithm as proposed by Chen et al [Chen et al., 2001]. The pixel disparity of two images can then be easily calculated from such a stereo matched image which is then used to reconstruct the 3D point cloud and the geometry of the scene. As moderately acceptable accuracy is sufficient for this work, the entire pipeline for

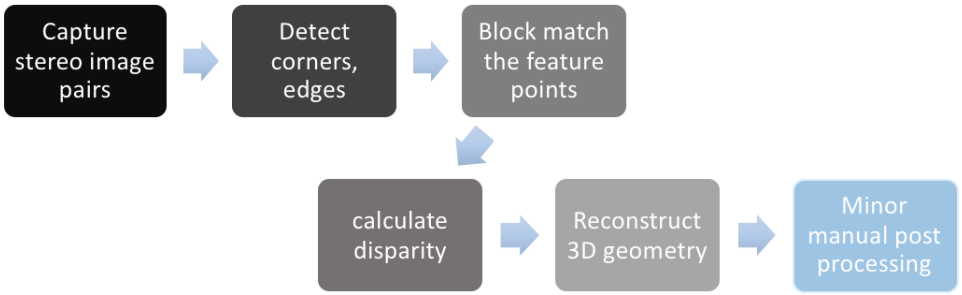


Figure 5.5: The scene geometry acquisition flow chart.

geometry detection has been implemented with already established methods and programming libraries. A nominal amount of manual post processing in the end produces a fairly usable scene geometry with minimal resources and effort.

5.2.1.2 Light Source detection

Although DCP can work with a high level of accuracy even with low resolution geometry, detecting the position and extent of the light sources accurately in the 3D scene is crucial for it and even for the ILF technique in general. Mapping the data into the appropriate data structures (in this case, SLF structures) is useful for changing the properties like position, colour and intensity. Unger et al. calculated SLF positions as a post processing of the ILF data with a semi-manual technique [Unger, 2009]. The method described here calculates the position of the light sources automatically, without any manual efforts, directly from the light probes captured in the scene. This method uses a minimum of two light probes captured underneath a physical light source in the scene. As ILF capture requires capturing multiple (thousands of) light probes around the scene, this method can detect the light sources with negligible additional effort and thus have been preferred than Unger et al's method in this work.

In the schematic diagram in Figure 5.6, one probe (P1) is captured underneath the light source. Several adjacent light probes are captured at a certain distance from the first probe. Intuitively, increasing the number of light probes will result in increased accuracy of the position and size detection. This hypothesis is later supported by the accuracy evaluation results in Section 5.2.1.3.

Thresholding of the light sources The detection algorithm samples the light source and generates a threshold value to distinguish the light source from the background. The extent of the light source is detected using the threshold value by a sliding window contrast detection method. A bounding box is created around the

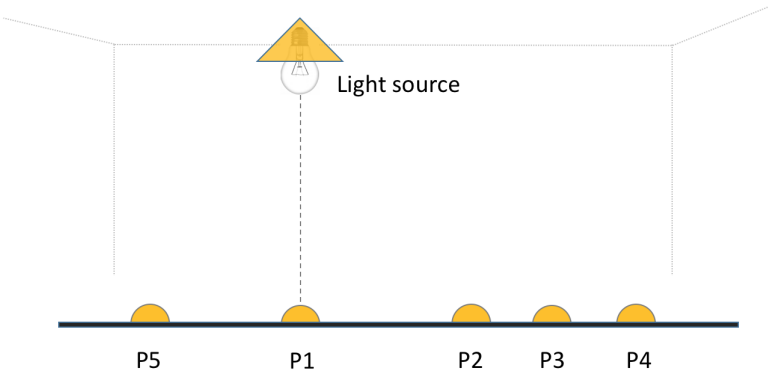


Figure 5.6: Thresholding and drawing a bounding box around light source in a real-world light probe image. The midpoint is detected as the pink dot.

light source using the extent values and subsequently the size of the box and middle point of it is detected. The direction $\vec{\omega}_1$ towards this midpoint of the light source is then calculated. As more subsequent light probes (P_i) are captured after this, the respective directions $\vec{\omega}$ can be used to determine the position of the light source.

Figure 5.7 shows the detected light sources in a light probe taken form one of the test real-world environment captures for this work. It should be noted that the straight fluorescent lights in the image has been warped due to the nature of light probe that renders the ($360° \times 180°$) environment and thus, the green bounding boxes around them do not cover their true surface area which leads to erroneous size estimation. As the light probe images are essentially warped into a hemispherical space and not planer images, it is helpful, if at least one of the light probes that has been used to detect the position has the light source in the center, i.e. a light probe right beneath the light source. This ensures the least amount of warping in the image and the size of the light source can be detected with greater accuracy.

Detection Methodology: Figure 5.8 shows the detection process in a 2D space. Two adjacent light probes has positions p_1 and p_2 respectively. The angles to the centre of the light source (**A**) for both of these probes can be calculated as position vectors $\vec{\omega}_1$ and $\vec{\omega}_2$ respectively. Here, the point p_1 and $\vec{\omega}_1$ makes the ray R_1 and the points p_2 and $\vec{\omega}_2$ makes the ray R_2. In parametric form, these are:

$$\begin{aligned} R_1(t_1) &= p_1 + t_1 * \vec{\omega}_1 \\ R_2(t_2) &= p_2 + t_2 * \vec{\omega}_2 \end{aligned} \tag{5.1}$$

Following figure 5.8, the rays R_1 and R_2 intersect at point A. The position of A can be found by solving the above mentioned linear equation 5.1. The point A is

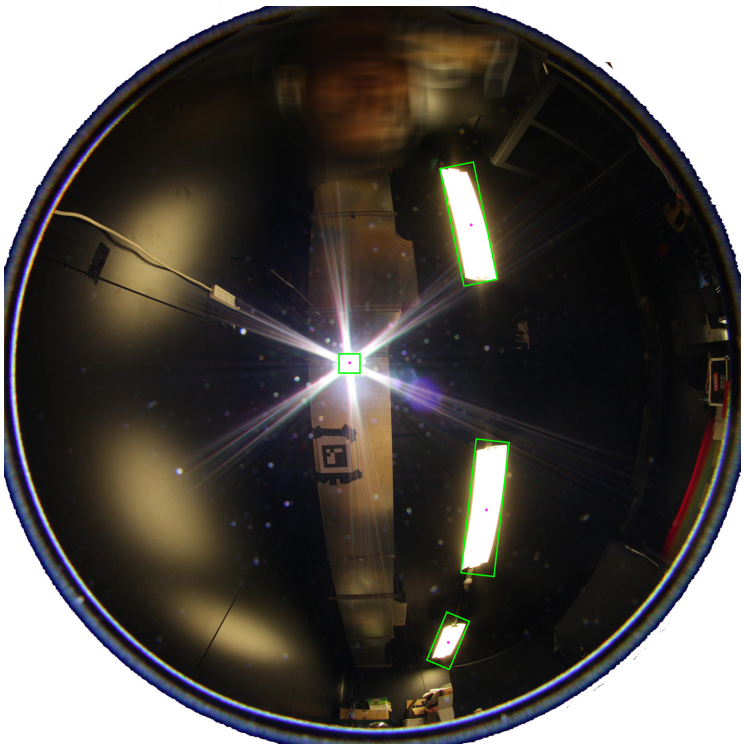


Figure 5.7: Thresholding and drawing a bounding box around light source in a real-world light probe image. The midpoint is detected as the pink dot.

the midpoint earlier shown in Figure 5.7 as pink dots in the light probe image. The dimension of the bounding boxes around the lights are then used to find the actual size of the light source subsequently.

Finally, The size of the bounding box and the calculated position co-ordinates are passed into the ILF preprocessor to create the SLF for the particular light source in the ILF. The next section evaluates the accuracy of this method.

5.2.1.3 Position Detection Accuracy Testing

The algorithm has been evaluated with a virtual scene which is a Cornell Box with a light source positioned at precise positions (x=0, z=0, y=5.5 m). The advantage of using a virtual environment is the ease of placing light sources of exact size at the exact position in the scene. The primary light probe is captured underneath the light source and 12 different secondary light probes at different distances from the primary light probe has been captured additionally.

Light probes of size 1000×1000 pixels have been used for testing. Figure 5.9 shows that the error percentage for position detection reduces sharply as the distance is increased between the light sources. This can be attributed to the fact that increased distance means better resolution of the angles with a finite resolution of the light probes. For real-world applications, 5% error in position detection is

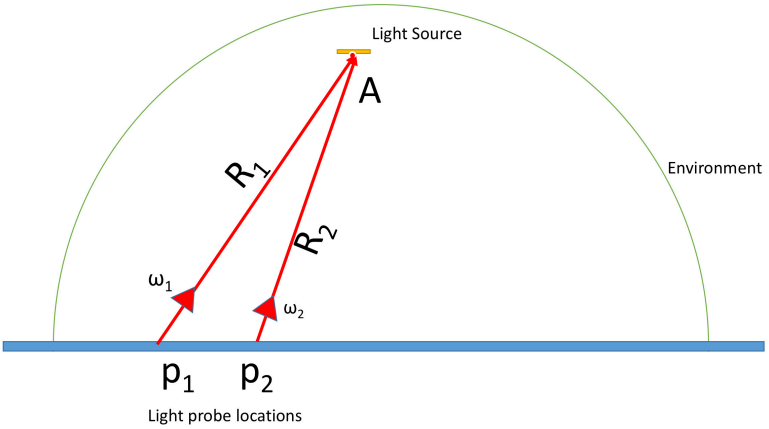


Figure 5.8: Position detection for a Light source by calculating the angle from two adjacent light probes to the middle of the light source.

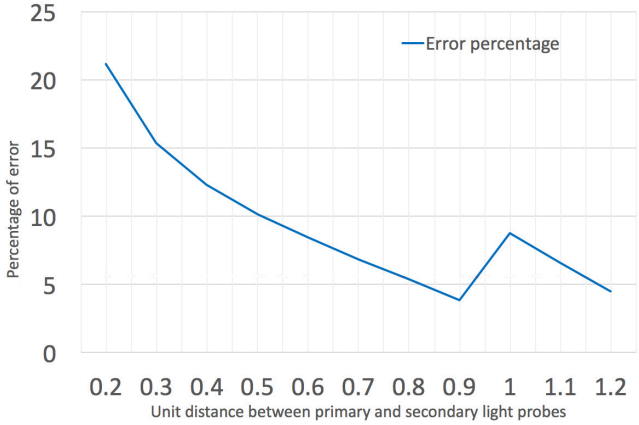


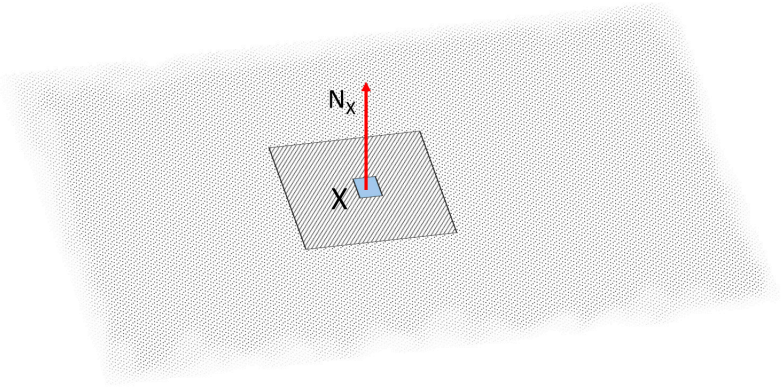
Figure 5.9: Error percentage vs. distance between primary and secondary light probes

negligible to the final rendering. It is noteworthy that there is a sudden peak around after the 0.9 unit distance mark. Most possibly this was due to the warping of the light sources previously seen in Figure 5.7 as the distances are increased. This also indicates that while it is better including many samples for the detection, it is advisable to not include samples which are too far from the light source to cause warping in the light probe images.

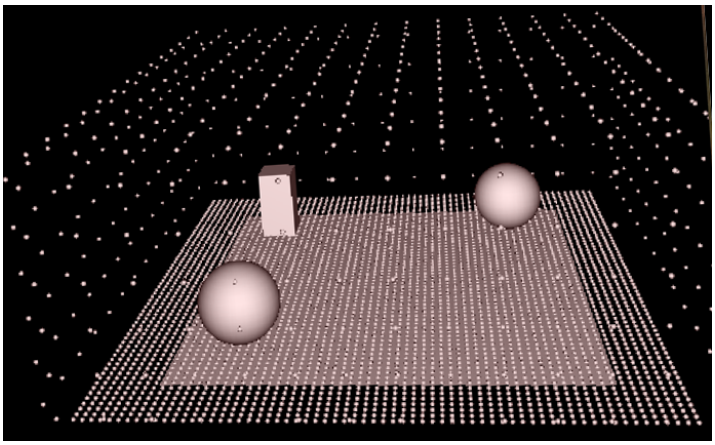
The next Section will describe the Update Points also known as UPoints, their distribution and the working principle in detail.

5.2.2 Update Points

Section 5.2 has described that the central working principle of the DCP algorithm is to estimate incoming light in various parts of the geometry. Update Points (UPoints) are imaginary patches of small (size varies with density) square areas which can be



(a) Structure of a UPoint. X is a UPoint on the geometry. The surrounding area patch of it is 1 square unit. NX is the surface normal similar to the area of the geometry it is placed on.



(b) The uniform distribution of UPoints around a simple scene geometry. Each small dots are UPoints. Note: higher density of UPoints along the floor.

Figure 5.10: The Update Points: Distribution and structure

distributed in any way (uniformly or randomly) along the geometry of the captured environment to estimate the amount of light coming to the corresponding parts of the scene. Figure 5.10b shows a uniform distribution of such UPoints around a scene as points. The UPoint distribution can be varied according to the requirements. This Section will discuss UPoint creation and various methods of distribution.

5.2.2.1 UPoint Structure

Each of the UPoints have their respective global position, the surface normal, the BRDF and the base colour properties. Each UPoint also has an unit area patch around them which is essential for variable density sampling and facilitating the solid angle parameter during the estimation of the incident illuminance amount as

described in Section 5.2.2.3. This section will go through the general structure and distribution of UPoints.

The surface normal of the UPoints are the same as the surface normal of the corresponding geometry of the environment. For the purposes of this work, all UPoint BRDFs are assumed as diffuse BRDF, thus reflecting light evenly. This can later be changed to any appropriate BRDFs assuming the corresponding surface in the real world is known. The base colour of all UPoints are also inherited from the corresponding area in the actual environment from the spatial ILF ray-database (ILF planes). The following Algorithm 2 describes the creation of UPoints and a uniform way to distribute them throughout the geometry.

Algorithm 2 Generating Update Points

procedure GENERATEUPOINTS
  $N_s \leftarrow$ number of faces in geometry
  $i \leftarrow 0$
  while $i \neq N_s$ **do**
    $N_a \leftarrow$ area of face F_i
    if F_i is $floor$ **then**
      $c \leftarrow \rho_{min}$
    else
      $c \leftarrow \rho_{min} * 10$
    end if
    $n = N_a * c$
    $j \leftarrow 0$
    while $j \neq n$ **do**
      assign memory to updatePoint UPt_{ij}
      $UPt_{ij}.position \leftarrow F_{i,j}.position$
      $UPt_{ij}.BRDF \leftarrow F_{i,j}.BRDF$
      $UPt_{ij}.sNormal \leftarrow F_{i,j}.sNormal$
      $UPt_{ij}.baseColour \leftarrow F_{i,j}.averageColour$
      $j := j + 1$
    end while
    $i := i + 1$
  end while
end procedure

5.2.2.2 UPoint Distribution

Algorithm 2 refers to a UPoint generation technique where the UPoints are distributed uniformly across the scene geometry as shown in Figure 5.10b. In case of

the floor, there are more UPoints per square unit area than the other parts of the geometry. The reason is, in tat particular scene, the floor reflects the light directly from a light source and thus is a "primary reflecting surface". The UPoint sampling density on the primary reflecting surfaces needs to be equal to the *minimum UPoint resolution* ρ_{min} calculation of which is described next.

Minimum Resolution of UPoints: Previously it was mentioned that UPoints have more sampling density around the "areas of interest" for increasing the accuracy of DCP in those areas. The primary reflecting surface is the first surface which receives the most amount of light directly from the light source. These surfaces are currently determined manually by the user.

The minimum resolution ρ_{min} of UPoints on a primary reflecting surface such as the floor depends on the size of the room and the just noticeable difference (JND) of the reflected light from the environment in case the light source changes position. Further exploring the idea, the JND is defined by the change of the intensity (ΔL_A) from an initial intensity (L_A) that is just noticeable to the human eye [Ferwerda et al., 1996; Larson et al., 1997]. In order to change the reflected light from the scene geometry to the order of JND, the light source needs to be moved a certain distance. The inverse square law of intensity and distance can calculate the required change in the light source position to make a just noticeable difference in the indirect light. Figure 5.11 describes the situation. The initial position of the light is at L1.

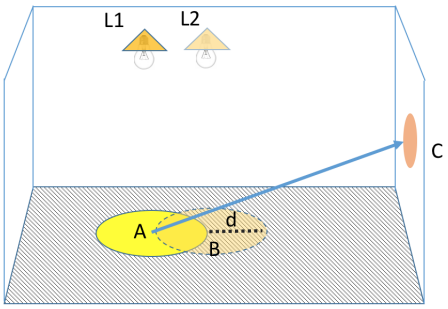


Figure 5.11: L1 is direct light position. Light is reflecting from L1 via the floor region A to the wall at C. To make a JND in intensity at C, the light needs to move to position L2.

The primary reflecting surface is the floor as the light reflects form the Region A to the region C in the wall. To make a just noticeable difference in the indirect light intensity from C, the light is moved from position L1 to position L2. Here, $L2 - L1 = d$ and d is the minimum resolution ρ_{min} of UPoints on the floor because if UPoints are spaced more than ρ_{min}, they will be unable to register the change in position as the light source is moved the least possible amount.

Minimum Resolution Calculation: According to studies on the perception capabilities of the human eye [Ferwerda et al., 1996; Larson et al., 1997] the JND ΔL_A

can be calculated with the following relations, given a initial luminance L_A:

$$log(\Delta L_A) = \begin{cases} log(L_A) - 0.395, & \text{if } -1.44 \leq log(L_A) < -0.0184 \\ (0.249 * log(L_A) + 0.65)^{2.7} - 0.72, & \text{if } -0.0184 \leq log(L_A) < 1.9 \\ log(L_A) - 1.255, & \text{if } log(L_A) \geq 1.9 \end{cases} \tag{5.2}$$

From equation 5.2 in case $0.96cd/m^2 \geq L_A \geq 10cd/m^2$, the JND is between $10 - 11.4\%$. The JND is around 10% where $L_A = 10 - 50cd/m^2$. For higher L_A, where $79cd/m^2 \geq L_A \geq 50cd/m^2$ the JND is around 8%. Since in practice the indirect light reflected from the scene geometry is well under $10cd/m^2$, the JND can be assumed as 10%. Assuming the initial and changed distance is $d_1, d_2, (d_2 - d_1) = d = \rho_{min}$ and the initial and changed intensities are I_1, I_2, the inverse square law yields:

$$\begin{aligned} \frac{d_1^2}{d_2^2} &= \frac{I_2}{I_1} \\ \frac{d_1^2}{d_2^2} \sim 1 &= \frac{I_2 \sim I_1}{I_1} \\ \frac{d_1^2}{d_2^2} &= 1 + 0.10 \\ d_2/d_1 &= 0.9534 \\ d &= 0.0466 * d_1 && [(d2 \sim d_1) = d] \\ \rho_{min} &\approx 0.05 * d_1 && [\rho_{min} = d] \end{aligned} \tag{5.3}$$

Equation 5.3 derives that given a length d_1 the ρ_{min} is 5% of d_1. In practice, d_1 usually is the biggest dimension of the scene geometry. Ideally the UPoint distribution should be following the ρ_{min} everywhere in the geometry. However, this can be computationally expensive and it is possible to reduce the number of UPoints to 25 times less by increasing the minimum resolution ρ_{min} by 5 times for all the other surfaces in the scene geometry that are not the primary reflecting surface. A simple bi-linear interpolation in the end as a post processing will produce smoother transition of the incoming radiance estimated on the area around UPoints.

Different UPoint Distributions: Although in this work, UPoints have been distributed uniformly across the scene geometry, different ways to distribute the points are still possible. Quasi-random functions such as Hammersley sequence [Wong et al., 1997] , Sobol Sequence [Kollig and Keller, 2002] or higher dimensional Halton sequences[Braaten and Weller, 1979] can be some of the ideal candidates for UPoint distribution. However, random sampling is useful for indeterministic scenarios, such as the Monte-carlo. DCP in the other hand requires defining an area in the geometry deterministically, thus deterministic uniform sampling fitting the geometry of the scene is preferable and more effective than random sampling.

5.2.2.3 Role of the UPoint Patches

As mentioned in the previous section and shown in the Figure 5.10a, the UPoints have an imaginary unit square surface area around it called the UPoint patch. The utility of this patch is two-fold. First, as Upoints are not randomly distributed and their distribution can vary according to the part of the geometry (for instance a high concentration along the primary reflecting surfaces), thus facilitating a more accurate technique to calculate the change in background ILF. A high density of UPoints leads to smaller UPoint patches, thereby compensating the UPoint density change in particular areas.

The second utility of the operating patches is about the amount of incoming light relative to the distance of the light source. As Figure 5.12 shows, areas closer to the light source received more light than the areas that are farther from it. The light coming to a UPoint is calculated by shooting a single ray to the light source (or the other UPoints for indirect light) from one UPoint and calculating the illuminance on the UPoint patch. A single ray carries the same radiance regardless of the distances. The amount of incoming light received by the UPoints should however be inversely proportional to their distance from the light source. (Figure 5.12).

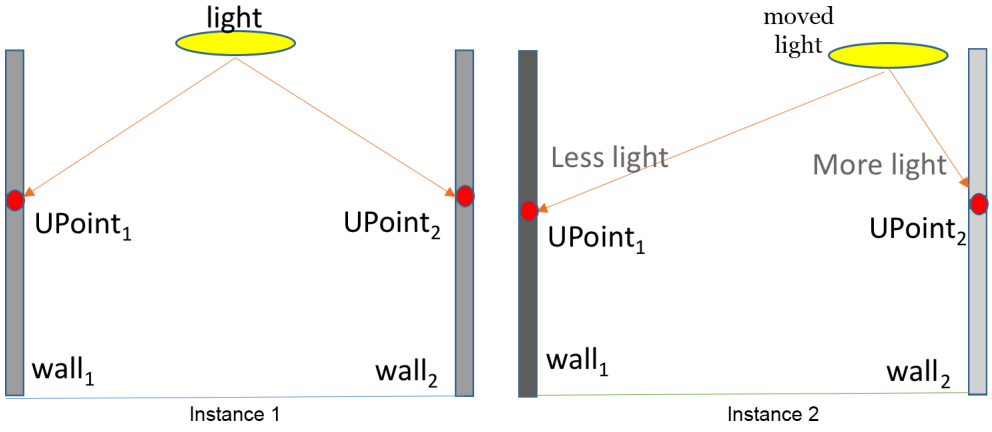


Figure 5.12: Two instances of same scene. The SLF has been moved in the instance 2 and thus the right side $wall_2$ has more light reflected from it while opposite $wall_1$ is darker.

In ray-tracing, this is achieved by sampling the light over an area where the accuracy logarithmically increases with the number of rays used [Dutre et al., 2006]. This can be an expensive process and a good estimation with random sampling of rays can take a significant amount of time. The DCP technique however needs to be fast since actual rendering with ILF can start only after the completion of the incoming light estimation and change propagation with DCP.

The solution proposed here treats the whole unit patch around each of these UPoints as the light source rather treating UPoints as point light sources. The rela-

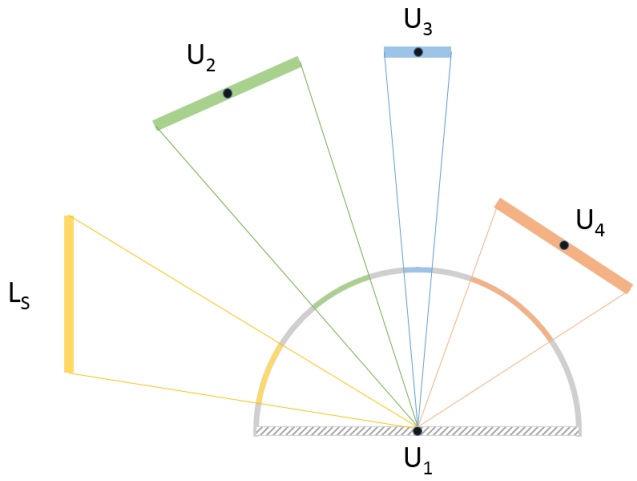


Figure 5.13: Incoming illuminance into UPoint U_1 from three different UPoints of different sizes and distances and a light source. The solid angle created by the UPoint patches are shown in different colour patches around the half circle encompassing U_1.

tionship between the amount of light received on a point and the distance, solid angle and the area of the light source is well established. Figure 5.13 further elucidates this. For example, UPoint U_1 in the diagram is receiving light from a light source L_s, and three other UPoints U_2, U_3, and U_4. Now, the distances from the various sources are d_{1l}, d_{12}, d_{13} and d_{14} respectively. The areas of the light source and the UPoint patches are a_l, a_1, a_2, a_3, and a_4 respectively. In this example, $a_l = a_2 = a_4$ and $a_l, a_2, a_4 > a_3$ while $d_{1l} = d_{12} = d_{13}$ and $d_{1l}, d_{12}, d_{13} < d_{14}$. For simplicity, the situation has been rendered in 2D. The half sphere around U_1 is to show the solid angle that is created with each patch. The different colour patches depicting the solid angle on the half circle encompassing U_1 elucidates the relations between the luminance contribution of a source with its distance and its area. A larger area with equal distance will result in bigger solid angle thus, more contribution; for example, the green and red patches. Similarly equal area with smaller distance will result in bigger solid angle, thus bigger contribution; i.e. the green and the red patches. The light source L_S however has the same area and the same distance with the UPoint U_2. This notably results in same solid angle, i.e. the yellow patch and the green patch. Thus, assuming same intensity, they have the same contribution on U_1.

The next section will describe the entire radiance estimation process as well as the necessary parameters used in detail.

5.2.3 Incoming Radiance Estimation

As mentioned earlier, the primary concept of DCP is to estimate the incoming radiance in the various parts of the geometry. The estimation is done by calculating incoming radiance on the UPoints distributed along the scene geometry. For *direct light* calculation on a UPoint, a single ray is shot towards each of the light sources from the UPoint and the radiance is stored in the UPoint data structure. This process is repeated for all the UPoints in the scene.

Subsequently, the *indirect light* contribution on a UPoint is calculated by shooting rays similarly from the UPoints, but to the other UPoints rather than the light sources. Repeating this process for a number of bounces estimates the radiance in all of these UPoints with a sufficient accuracy. This Section will now describe the procedure of estimating the incoming light, all the necessary parameters and the algorithm in detail.

5.2.3.1 Parameters for Radiance Estimation

As mentioned earlier, a single ray delivers only the irradiance at one point. Estimating the radiance over an area surrounding that point needs a few post processing on the UPoints. The radiance estimation module of DCP incorporates three parameters for this; (i) Cosine-term, (ii) Distance, and (iii) Solid angle.

(i) The Cosine-Term Parameter: According to the simplified rendering equation [Kajiya, 1986] the outgoing radiance L_o from a point X towards an outgoing direction is:

$$L_o = \int_{\Omega} fr(x,\omega) * L_i(x,\omega) * cos\theta * d\omega \tag{5.4}$$

Here L_i is the incoming radiance towards the direction ω. $f_r(x,\omega)$ is the BRDF component on point x and direction ω. The θ is the angle between the outgoing direction and the surface normal on point x. Figure 5.14 elucidates equation 5.4.

Assuming Point x is a UPoint and further assuming uniform diffused BRDF (fr) on x, the outgoing radiance from point X to point C is $L_o = fr * \Sigma L_i * cos\theta * d\omega$. Here, the total light (L_x) coming from x (towards the direction of the surface normal $\hat{N_x}$ is the total incoming light ΣL_i of point x multiplied by the BRDF fr and the solid angle $d\omega$. As the angle between $\hat{N_x}$ and $\bar{xc}$ is θ, the outgoing light from x to c is:

$$L_{x \leftarrow c} = L_x * cos\theta * d\omega \tag{5.5}$$

As the total outgoing radiance L_x is known, the $cos\theta$ is the first parameter of the radiance re-calculation module of DCP to calculate amount of light coming form one

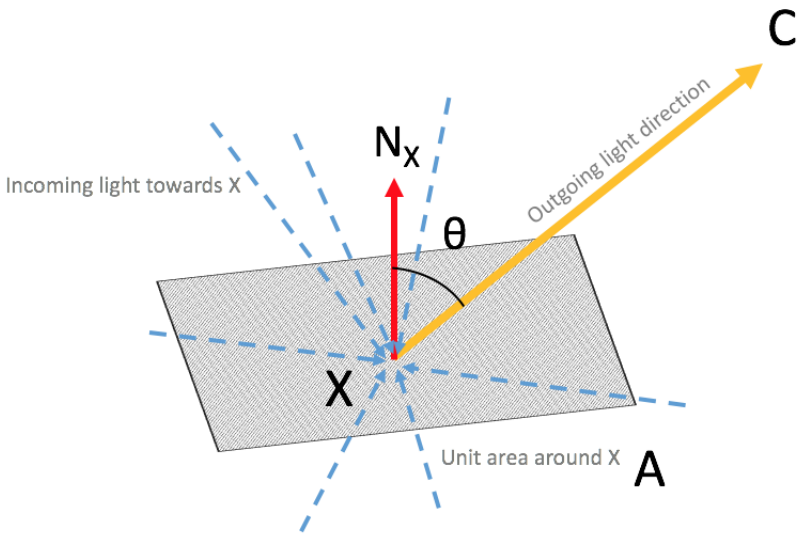


Figure 5.14: The relations of various parameters of the rendering equation.

source point (whether light source or a UPoint) to another UPoint.

(ii) The Distance Parameter: The effect of distance has been discussed in detail in Section 5.2.2.3 with the help of the example in Figure 5.12. The distance parameter is not used seperately in the estimation algorithm because is it already included in the solid angle parameter described next. The inverse square law between intensity (I) and distance (d) is:

$$I = k\frac{1}{d^2} \tag{5.6}$$

Where k is a constant that depends on the medium irregularities and other external influences [Dutre et al., 2006].

(iii) The Solid Angle Parameter: Section 5.2.2.3 describes the importance of the solid angle in detail during the explanation of the utility of the UPoint patches. Simply put, bigger solid angle means bigger area covered into the effective hemisphere of a UPoint which translates to more light contribution. The solid angle depends on the distance, the area and the orientation of a plane in front of it.

Calculating the solid angle is done with a known method. As shown in Figure 5.15, the area in front of the encompassing hemisphere around the UPoint is projected onto the sphere. This can be achieved simply by connecting four straight lines to the area patch and note the angles $(\theta_1, \theta_2, \phi_1, \phi_2)$. The solid angle ω is defined as the area on the surface of an unit sphere with altitude boundaries (θ_1, θ_2) and azimuth boundaries (ϕ_1, ϕ_2).

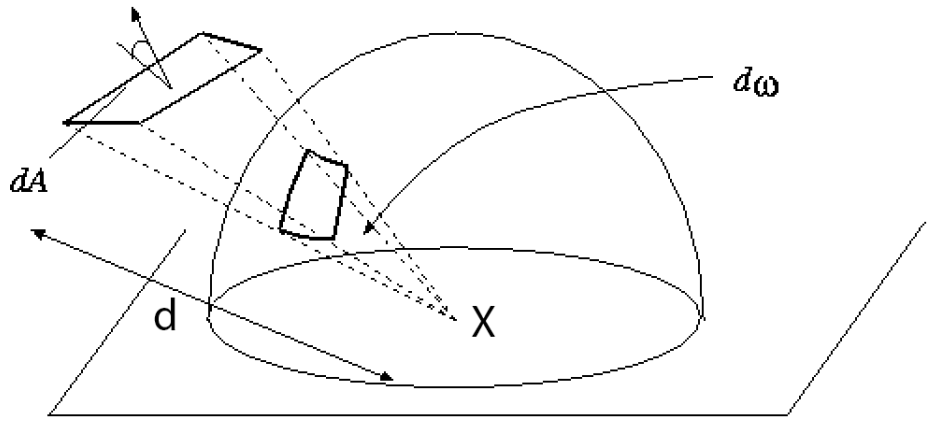


Figure 5.15: The solid angle formed by an area next to the encompassing hemisphere.

$$\int_{\phi_1}^{\phi_2} \int_{\theta_1}^{\theta_2} sin\theta d\theta d\phi = (\phi_2 - \phi_1)(cos\theta_1 - cos\theta_2) \tag{5.7}$$

Evidently, this technique of calculating the solid angle takes account the size, the distance and the orientation of the area patch in front of the concerning UPoint. As all the necessary parameters has been discussed, next Section will describe the radiance estimation algorithm.

5.2.3.2 Radiance Estimation Algorithm

The *generateUpdatePoints*() procedure described in Algorithm 1 generates the update points for different parts of the geometry with variable density. The radiance estimation is done on each of the UPoints. As the UPoints are located along the scene geometry, they acquire the basic properties of the part of the scene geometry to which they belong, as mentioned in Section 5.2.2.

The radiance estimation algorithm calculates the direct light contribution from the light sources on every UPoints in the first iteration. The indirect light contributions are calculated on three bounces on each of the UPoints coming from all the other UPoints. The algorithm 3 below calculates the estimated incoming light on all of the UPoints in the geometry.

The estimation algorithm uses ray tracing, as mentioned in Section 5.2.2.3, to calculate the direct lighting from the SLFs. There are two choices for the rays to calculate the light regarding whether to take account the occlusion by the model being rendered or not. For instance, the ILF is captured in a scene which does not include any occluders in between the capture area. The only boundaries of

Algorithm 3 Radiance Estimation Algorithm (part 1)

procedure CALCULATEINCOMINGRADIANCE
  $N_t \leftarrow$ *total* number of UPt
  $N_w \leftarrow$ *number* of UPt except floor
  $i \leftarrow 0$
  while $i < N_t$ **do** ▷ Direct Lighting
    $Pt_a := RandomSamplepointOnSLF$
    $Pt_b \leftarrow updatepointUPt_i.position$
    $r \leftarrow makeRayPt_btoPt_a$
    $col \leftarrow ILFevaluate(Pt_b, r)$
    $sAng \leftarrow calculateSolidAngle(Pt_b, Pt_a)$
    $ct \leftarrow calculateCosTerm(Pt_b, Pt_a)$
    $UPt_i.directCol \leftarrow UPt_i.baseCol * col * sAng * ct$
    $i := i + 1$
  end while ▷ Continued on next page.

the capture area is the boundaries of the original scene itself. Figure 5.16 further elucidates this situational example. The ILF has been captured in the Scene A which does not have any objects in the capture area to occlude the environment form the camera. Scene B can be thought as the rendered reconstruction of scene A with a synthetic object in the middle.

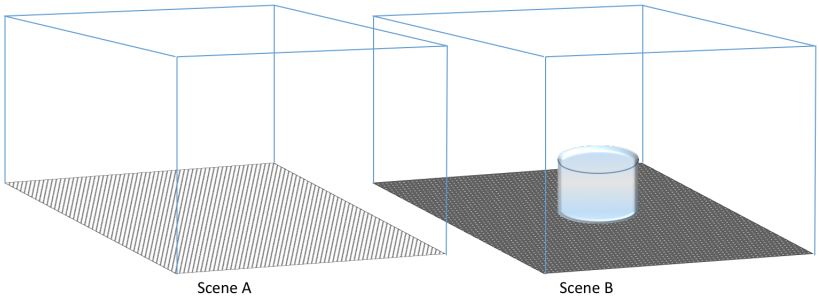


Figure 5.16: The ILF capture was done in Scene A without any object to occlude the entire environment while capturing. Scene B is the synthetic reproduction of Scene A with a synthetic object in the middle.

The DCP works during the rendering of an ILF, thus the synthetic objects that are being rendered is present in the scene. While estimating the incoming light on the UPoints, DCP has the choice of ignoring the synthetic models being rendered so the rays that are shot to the light sources will not be occluded by the added synthetic objects. This way DCP can estimate the baseline incoming radiance L_b in the original scene. In order to estimate the changed incoming radiance L_c on the UPoints, DCP must take the occlusion caused by the newly added synthetic objects

Algorithm 4 Radiance Estimation Algorithm (part 2)

```
15:   bounce ← 0
16:   while bounce < 3 do                                   ▷ indirect lighting
17:       i ← 0
18:       while i < N_t do                                  ▷ N_w for speedup
19:           j ← 0
20:           while j < N_t do
21:               if UPt_i == UPt_j then
22:                   continue
23:               end if
24:               Pt_a ← updatepointUPt_i.position
25:               Pt_b ← updatepointUPt_j.position
26:               sAng ← calculateSolidAngle(Pt_b, Pt_a)
27:               ct ← calculateCosTerm(Pt_b, Pt_a)
28:               UPt_i.indirectCol += UPt_i.baseCol * UPt_j.directCol * sAng * ct
29:               j := j + 1
30:           end while
31:           i := i + 1
32:       end while
33:       bounce = bounce + 1
34:   end while
35:   i := 0
36:   while i < N_t do                                  ▷ Estimated incoming irradiance
37:       UPt_i.incomingRad ← UPt_i.directCol + UPt_i.inDirectCol
38:   end while
39: end procedure
```

so the change propagation is much more realistic and accurate to the scene being rendered.

The indirect lighting is calculated similarly, but instead of querying the light sources it uses the radiance stored from direct lighting in the UPoints. The iterations are kept to one direct and two indirect lighting (total 3 bounces) for the scenes tested. This can be increased for greater accuracy if needed at the cost of speed. The following section will describe the process to use this estimated incoming radiance on the UPoints to change the ILF database accordingly.

5.2.4 The Change Propagation

The estimated incoming radiance on the UPoints are used to update the subsequent parts of the ILF data. The UPoints are placed along the scene geometry, which

has a one-to-one mapping to the ILF database, thereby capable of retrieving the ILF data corresponding to any arbitrary part of the geometry. As mentioned in Section 4.3.1.2, ILF data can be stored in various different ways, all of which are based on spatial data structures. Originally this data structure does not need to correspond to the real geometry of the scene however mapping them to a geometry is a trivial task. Chapter 3 described the ILF re-projection method to map the ILF data into 2D planes encapsulating the capture space called the *ILF planes*. The DCP implementation in this work extensively uses the ILF plane data structure, albeit any other data structure and an appropriate mapping to the geometry will work similarly.

5.2.4.1 Updating the ILF

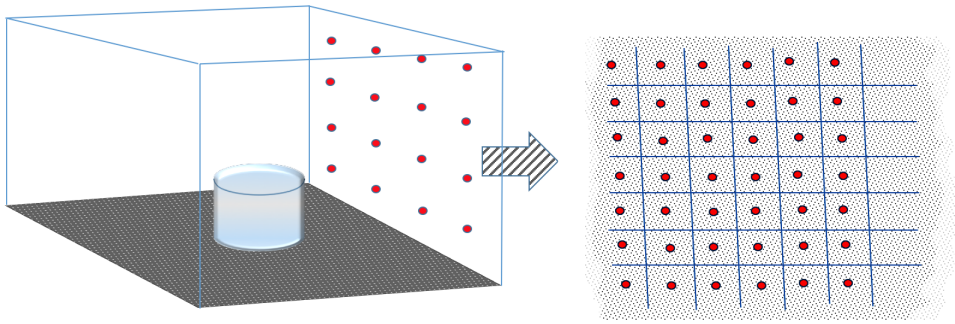


Figure 5.17: Schematic diagram of a reconstructed scene with ILF. The red dots are the UPoints. For brevity, only one wall of the geometry is showing the UPoint distribution. The right image is a expanded view of the UPoint distribution on the wall showing the spatial areas that each of them are entitled to update in the actual ILF database.

The UPoints are distributed uniformly along the geometry spatially. Each such UPoint has an operating area of the respective part of the geometry. This area is used to access the spatial data entries of the same area in the ILF database.

Each such area has two variables for the total incoming baseline radiance (L_b) and the updated total changed incoming radiance (L_c). This has been implemented in the UPoint structure itself. For a UPoint i, the L_b^i is measured and stored before the start of the rendering. Each time the light is changed, the total incoming radiance is estimated again and stored in (L_c^i).

Let's assume that an arbitrary point X in the environment belongs to the operational area of the UPoint i. The ILF plane corresponding to the UPoint i has the outgoing radiance information L_o from point X known. Let's further assume that, if the light in the environment changes, the outgoing radiance from X changes to L'_o, which is not known to the ILF. As X belongs to the UPoint i, the incoming

radiance to point X was calculated as L_b^i while the outgoing radiance from point X was L_o. After the change in the light source, the changed incoming radiance on X is calculated as (L_c^i) while the changed outgoing radiance from X, L'_o, is unknown. Now as the BRDFs has been universally assumed to be diffused, the ratio of (L_b^i/L_o) and the (L_c^i/L'_o) is same when all other conditions are unchanged. From this equivalency, we have:

$$
\begin{aligned}
(L_b^i/L_o) &= (L_c^i/L'_o) \\
L'_o &= L_o * (L_c^i/L_b^i) \\
L'_o &= L_o * k_i \qquad\qquad \text{where } k_i = (L_c^i/L_b^i)
\end{aligned}
\tag{5.8}
$$

The ratio $k_i = (L_c^i/L_b^i)$ (or $k = (L_c/L_b)$ by simplifying for all UPoints) is called the *Change Propagation Ratio* of a UPoint. Similarly, for any arbitrary point in the geometry, ILF database has the outgoing illuminance L_o. To obtain the changed outgoing radiance L'_o from that point, the known value of L_o is multiplied by the Change Propagation Ratio k (Equation 5.8) from the corresponding UPoint.

Currently the assumption of diffused BRDF on UPoints is a limitation of DCP technique. Work is going on to implement a more robust change propagation technique that accommodates other complex BRDFs as well.

During the actual rendering of a synthetic scene, each ray query to the ILF database requires a spatial position i and an angular position ω which in turn returns a colour value which is simply multiplied by the k_i to obtain the final updated illuminance. This update process is independent of the data structure used and can be adopted to any other implementations as long as there is a method to obtain the corresponding spatial position of the data which has been queried from the ILF.

The following Section 5.3 describes several tests regarding the various aspects of the DCP technique.

5.3 Evaluation of DCP Technique

DCP has been designed for real-world data, however for methodological and implementational reasons discussed next, it has been evaluated with virtual environments.

The reason behind using virtual environment instead of a real-world capture is twofold. *Firstly*, is easier to manipulate a virtual scene precisely. The lights can be positioned exactly with exact power. The material properties and the colour of the scene boundaries also could be precisely controlled to eliminate all possible variables while evaluating DCP.

Secondly, This work had access to only the static light probe setup described in 4 with which, real-world ILF captures are extremely time consuming. In less than 1 square meter area, Unger et al. captured around 50,000 light probe samples [Unger, 2009]. This could be possible for high precision expensive instruments, however, this work did not have access to such equipments. In comparison to the 50,000 sample capture, this work describes a similar real world implementation in the next chapter for Temporal ILF where it took 7 hours to capture just 1163 HDR light probes around a similar dimension space. Evaluations needed hundreds of different ILF capture sessions, each capturing multiple thousands of light probes. It would not have been feasible with the real-world ILF captures.

The accuracy of the DCP technique thus has been evaluated by capturing ILFs in a controlled environment where it is possible to precisely change the positional and radiometric properties in the direct light sources and have unlimited sample density of the captured light probes in a matter of few hours by directly rendering the virtual scenes with a "fisheye" lens camera shader developed in house for this purpose.

The evaluations has been done by comparing the indirect light stored in two different ILF captures, ILF_1 and ILF_2. ILF_1, which is used as the "ground truth", has a light source in a certain position and radiance (X_1, L_1) whereas in ILF_2, the light source has the position and radiance (X_2, L_2). Firstly, the light source in the ILF_2 is synthetically edited with the DCP technique to change its position to (X_1, L_1) from (X_2, L_2) in order to match the position and radiance of the ground truth. As the light source is changed, DCP changes the indirect light in the edited ILF accordingly which ideally should resemble the indirect light stored in the ground truth. Subsequently the indirect light stored in the ILF_1 and the edited ILF_2 is then compared to measure the level of accuracy of the DCP technique.

The comparisons have been done with two types of change propagations i.e. the position change and the change of the radiance of the light. Each of these comparisons has been conducted with two virtual environments; one with an even geometry and the other with several ridges added to the scene geometry making the boundaries uneven. While the first scene provides an ideal condition to evaluate the performance of DCP, the second environment is useful for comparing the effects of high resolution geometry which accurately covers the uneven scene boundaries, over low resolution geometry which assumes an averaged evened-out scene boundaries on DCP.

The following Section 5.3.1 describes the evaluation with simple virtual environment. Subsequently, Section 5.3.2 describes the evaluations with the virtual environment with uneven scene boundary.

5.3.1 Environment with Simple Geometry

As mentioned above, evaluations with real-world captures can be impractical and extremely time consuming. Controlling the power of the light source precisely to arbitrary values are also very difficult to implement in the real-world. To maintain control over the environment, the evaluation methodology of the DCP technique makes use of an ideal virtual environment with precisely controlled synthetic light sources.

The virtual scene is a simple room with $(14 \times 14 \times 6)$ unit dimensions. The four walls around it are coloured differently for detection convenience of colour shifts and colour bleeding (Figure 5.18). The light source is a spotlight with a 30° light cone. Several ILFs are captured using this environment with changed light source positions and radiances. For the DCP technique, this scene had 57000 UPoints spread across the scene geometry.

5.3.1.1 Capture

The ILF requires 360° light probes in the capturing phase which is essentially a HDR picture taken with a fish-eye lens and a planar camera. For the purposes of this experiment a fisheye camera shader has been developed and used to capture the synthetic light probes throughout these experiments. The capture space is a (10×10) unit rectangle at the center of the floor of the virtual environment. At each 0.2 unit distance, one image is taken totaling to 2500 HDR 360° light probes of the upper hemisphere.

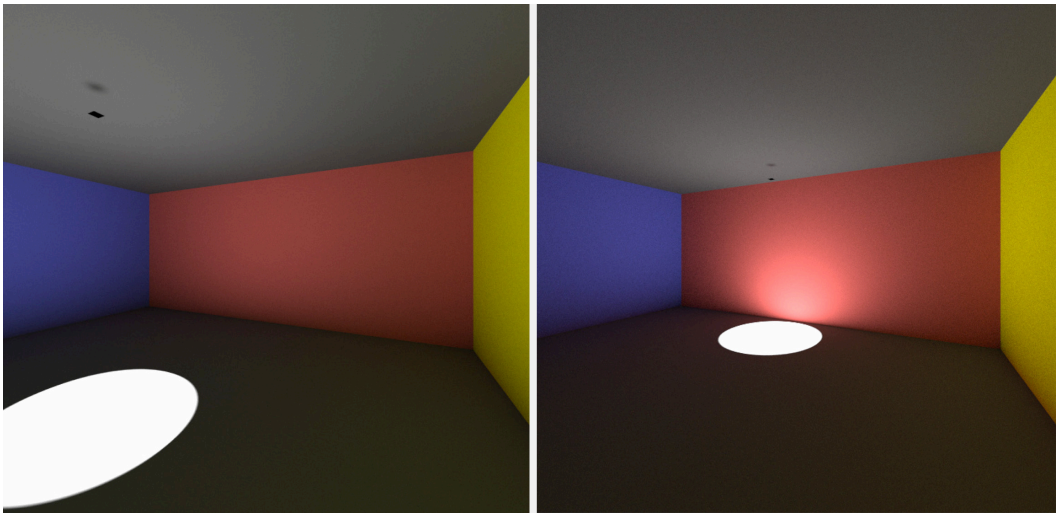


Figure 5.18: Virtual Environment with 2 spotlight positions. The walls in the scene are differently coloured to detect colour bleedings when the spotlight is pushed to the edge.

For the evaluation, 13 sets of ILF captures has been done with 8 different

positions of the light source as well as 5 different radiances. All these sets of ILF captures are considered as the ground truths for their specific spotlight position x and radiance L. Figure 5.18 shows the environment from two such ILF captures, one with the spotlight in the middle and the other with the spotlight 5 unit away from center, at the edge of the room. Each of the positions and radiances of the spotlight in these ground truth captures are compared with the the simulated change of the same position and radiance with the DCP technique.

5.3.1.2 Methodology

The evaluation is done by comparing the two sets of ILF capture; the ground truth $G_{1,1}$ with position x_1 and radiance L_1 and a simulated ILF $S_{0,0}$ where the original position and radiance of the light source was different (x_0, L_0) which has then been changed to match the position of the ground truth. Although the ILF capture used for simulating other positions and radiances can have any arbitrary (x_0, L_0), in practice the x_0 is in the center of the room (3D global position (0,0,0)) with radiance $L_0 = 1$.

As DCP is used to change the position and radiance from (x_0, E_0) to (x_1, E_1), the DCP algorithm propagates these changes to the entire ILF database, thus changing the indirect light reflecting from the geometry in the simulated ILF $S_{0,0}$. The indirect lights are then retrieved from the $S_{0,0}$ and the ground truth $G_{1,1}$ and are subsequently compared to obtain the error as a measure for the accuracy of the DCP technique. This method is repeated for several different ground truths with varying positions and intensities to get a distribution of errors against the amount of the change. The next section describes the results for the movement comparisons with different positions of the light source.

5.3.1.3 Position Change Propagation Evaluation

Propagating the changes in indirect lighting after movement of the light sources has been evaluated using eight different environments with varying light source positions at $X = 0, 2, 2.5, 3, 3.5, 4, 4.5, 5$ along the global X axis with the same radiance. In Figure 5.18 shows two such environments with light source at X=0 and the light source at $X = 5$.

As mentioned before, this experiment uses $X = 0$ position as the base ILF for the propagation. For example, if the light is at X=5 in the ground truth, then the SLF in the propagated case is moved from its base case $X = 0$ to the position of the ground truth at $X = 5$. The reflected indirect light data from both the cases are obtained from the respective ILFs and subsequently, the average reflected luminance is calculated. This process is repeated for 7 different positions of the light

source $X = 2, 2.5, 3, 3.5, 4, 4.5$ and 5. The standard RMSE errors provides a general measure of accuracy of the output at different position of the light sources.

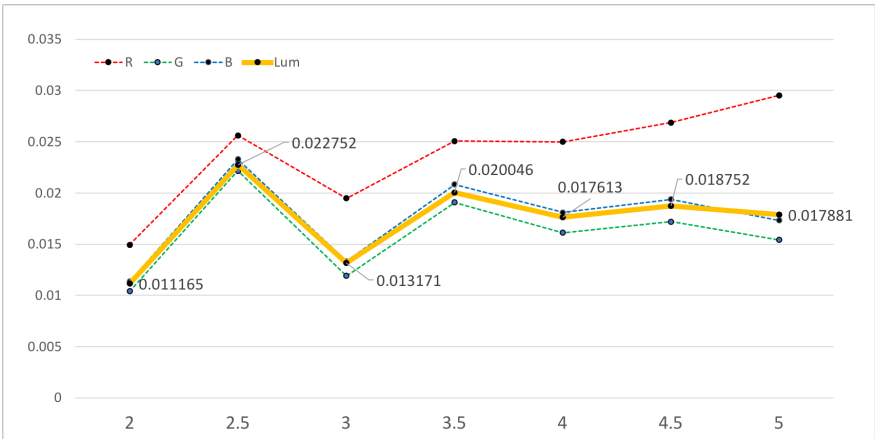


Figure 5.19: RMSE values for the indirect light from scene geometry. The X axis is the unit distances from the base position of the light source as it is moved towards the right side. Y axis is Normalised RMSE calculated from the entire geometry

The Root Mean-Squared Error (RMSE) values in the graphs shown in figure 5.19 has been calculated for the luminance at 8 different positions where the SLF were moved on the global X axis. The RMSE values are normalised by the range of the baseline values. As the light source has been moved gradually towards the right hand direction of the scene, the error values are calculated for indirect light bouncing off the scene geometry.

The graph at Figure 5.19 shows the error values increase with a positive trend. The normalised RMSE (nRMSE) has been calculated for each red, green and blue channels separately (dotted lines in graph) as well as the overall Luminance. The edge position at $X = 5$ however shows the highest error measures in the red channel whereas the overall luminance values remain close together, averaged at nRMSE 0.016.

The Absolute Percentage Error is the second type of error that has been calculated throughout the space span of the scene geometry; i.e. the $((V_{original} - V_{moved})/V_{original})$ ratio for the different areas of the captured geometry. The heat-map in Figure 5.20 shows the absolute error percentages at 84 (14×6) different parts of the right wall of the scene while the SLF was moved to the $X = 5$ position form $X = 0$. Here, the horizontal axis of the wall is spanning from 1 to 14 and the vertical axis (right side wall) and spanning from 1 to 6. The colour map on the right shows the absolute error percentages as a reference to the heat-map. This provides a clearer idea about the way the update propagation is affecting different parts of the geometry and provides pointers on ways with which the results can be made more accurate.

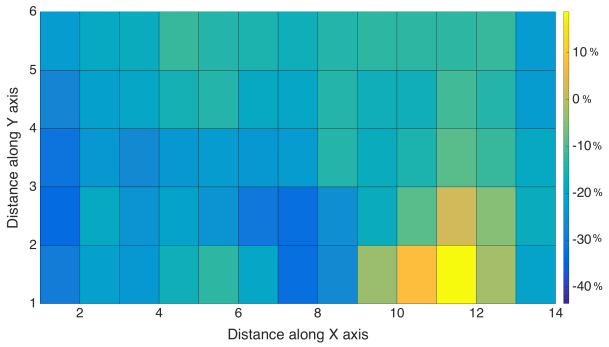


Figure 5.20: Absolute percentage error while light source moved to position X=5. This is a colour map of error in different part of the right side of the captured scene. The dip at bottom middle shows the original values were bigger than the recalculated ones.

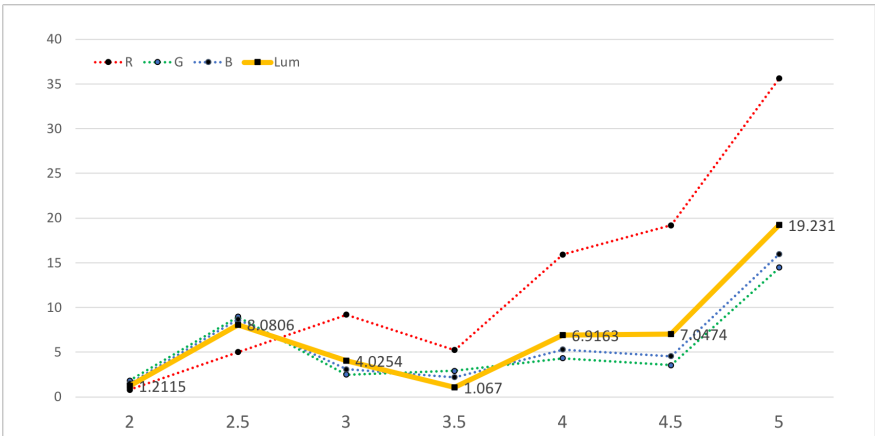


Figure 5.21: Absolute percentage error for movement of light source. The X axis is the unit distances from the base position as the light source is moved towards right side. The Y axis is the percentage of error calculated from the entire geometry.

The average absolute percentage error can be seen increasing in the graph described in Figure 5.21 as the light source is moved towards the edge of the scene geometry, complementing the graph in Figure 5.19 for the average RMSE values. The absolute percentage error at light source position $X = 5$ at the edge is the highest and the error in the red channel is higher since the base colour of the geometry was primarily red. The overall luminance error until before the edge has been quite low at 7.0%. When the light source is at the edge of the geometry, the luminance error is 19.23%. Considering this is almost direct lighting with the light is shining right into the wall with an angle, error percentage of 19.23% is still fairly acceptable given the JND is at 11% as calculated previously in Section 5.2.2.2.

One of the possible causes of this is the distribution pattern and density of the UPoints. A large number of UPoints could mitigate the spike in the percentage error in the geometry boundaries. In future, more distribution sequences could be

compared with each other in order to increase the accuracy even more. Another possibility is the inaccuracies of the assumed perfectly diffused BRDFs on the geometry. In future, a more thorough exploration of this could be helpful.

5.3.1.4 Radiance Change Propagation Evaluation

As previously mentioned, DCP technique has been tested with the change in the radiance of the light source and how the changes are propagated in the entire ILF database as well. Similar to the methodology described in the Section 5.3.2.3, different "ground truth" ILF captures has been done with varying radiances while the light source position is fixed in the middle of the scene. The radiance of 500 Cd/m^2 is used as the *"base case"* from which the changes are made to be compared against the 8 ground truths which are 10%, 25%, 50%, 75%, 250%, 500%, 750% and 1000% of the base radiance.

Similar to the Section 5.3.2.3, normalised RMSE values and the absolute percentage errors are calculated for the different percentages of the light source radiance. However, as the errors are negligible, the analysis of the space spanned

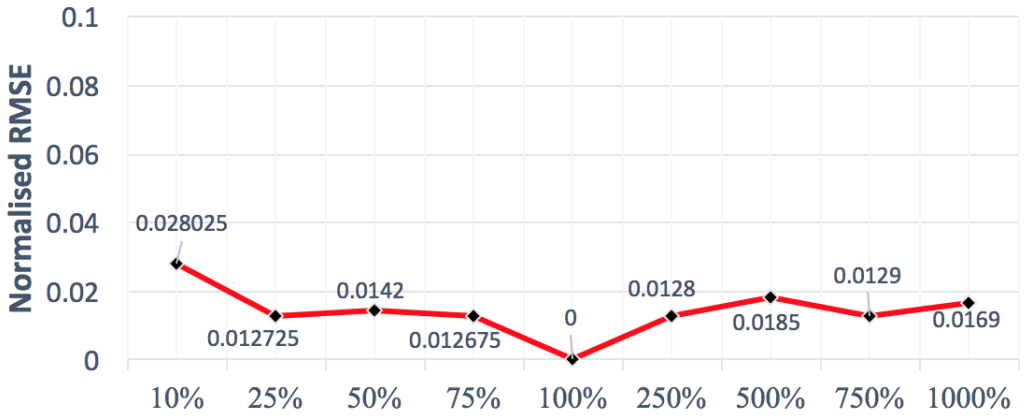


Figure 5.22: RMSE values according to the changed radiance. The X axis shows percentage of radiance in the light source and the Y axis shows the error values.

percentage error similar to the previous experiment of light movement propagation is not necessary in this case and thus omitted. The graph described in Figure 5.22 shows the range normalised RMSE in the Y axis while X axis is the percentage of the radiance which were changed for the DCP algorithm to propagate. The average percentage of error throughout the geometry as the changes were propagated are negligible as well with the highest level of error in some regions of the scene is 8% with an average of 3.41% which is unlikely to make any tangible difference.

5.3.1.5 Accuracy vs. Efficiency of the DCP Technique

The DCP technique calculates the estimated incoming radiance by a ray-tracing based approach which is repeated several times for indirect light contribution calculation among the UPoints which has been described before. For N number of UPoints, the first bounce requires N ray calculations and the subsequent bounces require N^2 making the complexity (upper bound) where B is the number of bounces:

$$O(B) = N + (B - 1) * N^2 \tag{5.9}$$

Clearly, the increment is linear as the number of bounces B is increased. This might not be a big problem for a small number of UPoints but can be an issue for a very large N, specially if an application requires real time changes.

Following this, a naive null hypothesis is, more bounces will result in greater accuracy and lesser errors. This evaluation recorded the computation time for increasing number of bounces (B) and subsequently recorded the average radiance from the updated ILFs for each of these bounces to compare them with the "Ground truth" data.

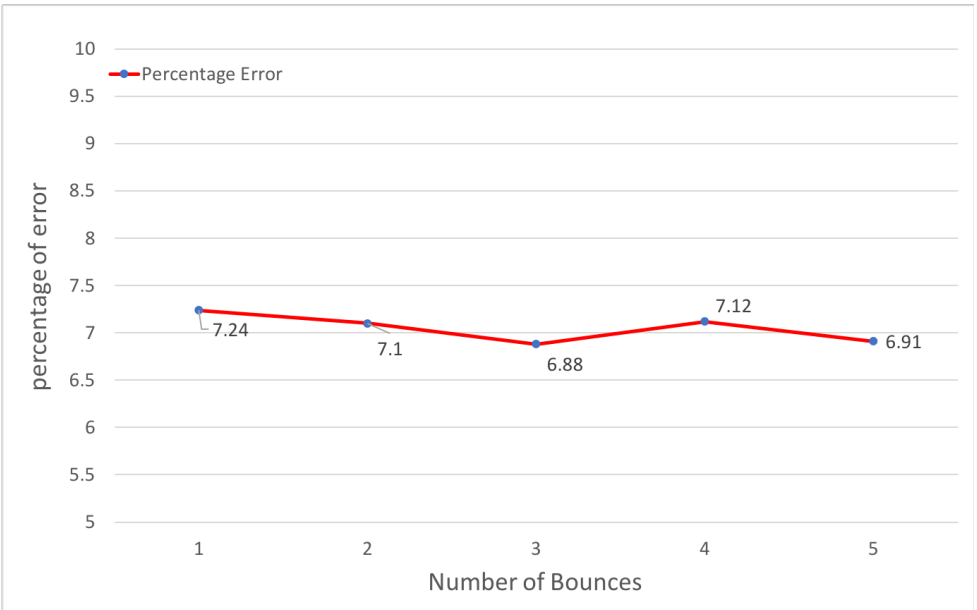
The experiment was conducted in a Intel Xeon workstation with 24 physical and 48 logical cores. The computation times for each bounce were recorded for 7 different positions and then averaged over them. The ground truth data were the same as the ones used in Section 5.3.2.3 for movement evaluation before. The percentage errors for each of these bounces are calculated with the similar procedure described before. Figure 5.23 shows the graph of the time elapsed and the percentage of errors for 6 increasing bounces starting from 3 to 8.

The data shows that while there is indeed a linear increment in the computation time as bounces are increased, there are no discernible trend of improvement of the errors after the 4th bounce. As the hypothesis is proved wrong for this test case, it can be concluded that increment in the bounces will not necessarily mean increased accuracy. Although more bounces can be implemented for scenes with smaller number (N) of UPoints; for larger N, it might not worth having the extra bounces sacrificing the speed.

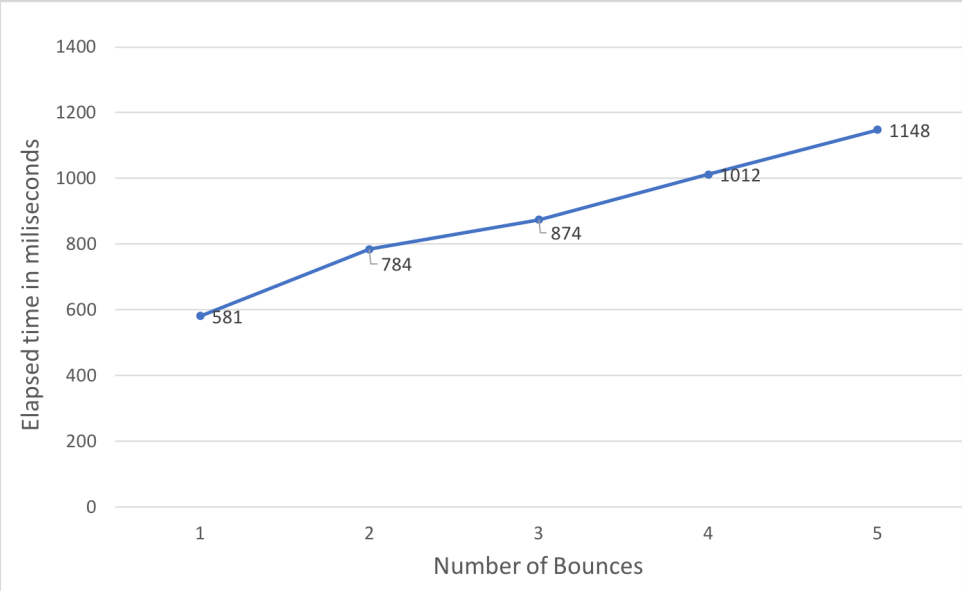
The next section will evaluate the DCP with an uneven geometry as the test environment as opposed to the simple geometry used in this section.

5.3.2 Environment with Uneven Geometry

The goal of evaluating DCP with an environment with uneven geometry is to explore the effects of an unevenness of the scene geometry on low resolution boundaries, i.e. in cases where the captured scene geometry information does not follow all the ridges and troughs in a scene geometry. This evaluation is important because it is



(a) Averaged absolute percentage Errors (7 movements of light source) with respect to UPoint Bounces. The X axis number of bounce and the Y axis shows the error values.



(b) Average computation time (averaged on 7 movements of light source) with respect to UPoint Bounces. The X axis number of bounce and the Y axis shows the time in miliseconds.

Figure 5.23: Accuracy vs Speed of DCP

not always possible to have high resolution laser scanenrs to acquire the accurate geometry of the scene. A crude geometry can be acquired much simply by SfM based techniques described previously in this chapter. If the error rates are similar for high and low resolution geometry, acquiring ILFs of a complex scene and applying DCP on it could be made possible with an acceptable degree of accuracy.

Figure 5.24 shows a schematic diagram of such a geometry. Here, the actual geometry of the room on the left side with two pillars and a ridge has been shown in a top-view in the right. A high resolution geometry will cover all the unevenness in the boundaries along the dark grey lines. The red dotted line boundary is the low resolution geometry boundary which does not cover all the unevenness meticulously and provides an average scene boundary.

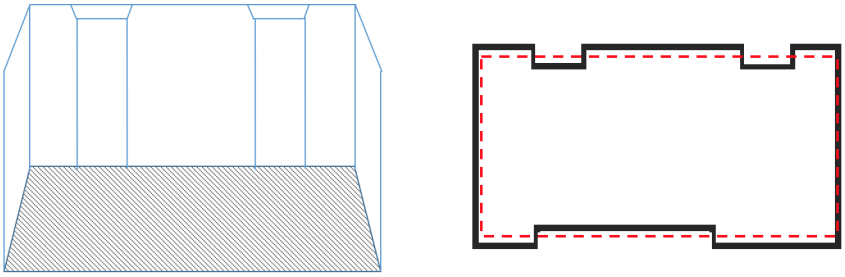


Figure 5.24: Schematic diagram of a room with uneven boundary. The right image is the top view of the boundary geometry of the room. The red dotted line is the low resolution boundary averaging the geometry.

Similar to the Section 5.3.1, the test environment with uneven geometry has been captured using a virtual environemnt. The virtual scene is a room with similar dimensions ($14 \times 14 \times 6$ units) although the walls around it are not uniform. There are pillars and ridges on the walls which are in similar proportions ($0.3 units$ in thickness, $0.6 units$ in width) to a physical room on which the virtual environment has been loosely modeled. Figure 5.25 shows two walls of the environment. The light source is the same spotlight with the 30° light cone.

5.3.2.1 Capture

The ILF captures has been done exactly similar to the previous section. All the ILF captures for the ground truths have been captured exactly same. The geometry however has been interpreted in two different resolutions, i.e. the accurate description of the actual uneven geometry and the average description of the geometry not taking the unevenness into account. This does not however affect the capture process in any way because the different geometry resolutions are only relevant for mapping the ILF ray-database to their corresponding geometry.

5.3.2.2 Methodology

The evaluation has again been done by comparing the ground truth $G_{1,1}$ with the simulated ILF $S_{0,0}$ where the light source has been changed to match the position of the ground truth. The "base case" for the simulation is identical to the previous

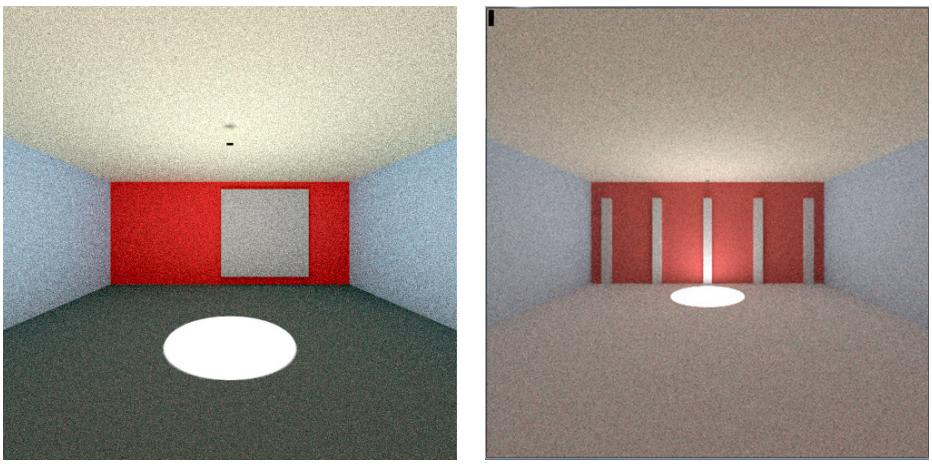


Figure 5.25: Virtual Environment with pillars and ridges contributing to an uneven boundary. Right side image has spotlight placed near the edge.

environment, in the center of the room (3D global position (0,0,0)) with radiance $L_0 = 1$.

The additional variation in the comparison comes with the variation in the geometry. The distribution of UPoints has been done in two ways; (i) which follows high resolution geometry boundaries, (ii) which follows a low resolution approximated geometry boundary (e.g. the red dotted line in Figure 5.24). On the basis of these two separate distribution of UPoints, the DCP is used to change the position and radiance from (x_0, L_0) to (x_1, L_1) and the the simulated indirect light is collected for both of these cases. The indirect lights are similarly retrieved from the $S_{0,0}$ as well as the ground truth $G_{1,1}$ which are subsequently compared to obtain the error as a measure for the accuracy of the DCP technique. This method is repeated for both the cases (high and low resolution boundaries) as well as repeated over several different ground truths with varying positions and radiances similar to the testing procedure in the previous section to get a distribution of errors against the amount of the change. The errors are then compared with the two cases to find out whether there is a significant amount of difference in them. The next section describes the results.

5.3.2.3 Position Change Propagation Evaluation

The light source movement evaluation has been done with the similar eight different environments with varying light source positions at $X = 0, 2, 2.5, 3, 3.5, 4, 4.5, 5$ along the global X axis with the same power of the light. Figure 5.25 shows two walls of the same environment used with light source at $X = 0$ and the light source at $X = 4.5$.

Similar to the previous evaluation, this experiment uses $X = 0$ position as the base ILF for the propagation. The reflected indirect light data from both cases are

obtained from the respective ILFs and subsequently, the average reflected luminance is calculated in the same procedure as the previous. This process is repeated for 7 different positions of the light source $X = 2, 2.5, 3, 3.5, 4, 4.5$ and 5 for each of the two cases (high and low resolution). Similar to the previous evaluation, normalised RMSE and the absolute percentage error has been calculated compared between both cases.

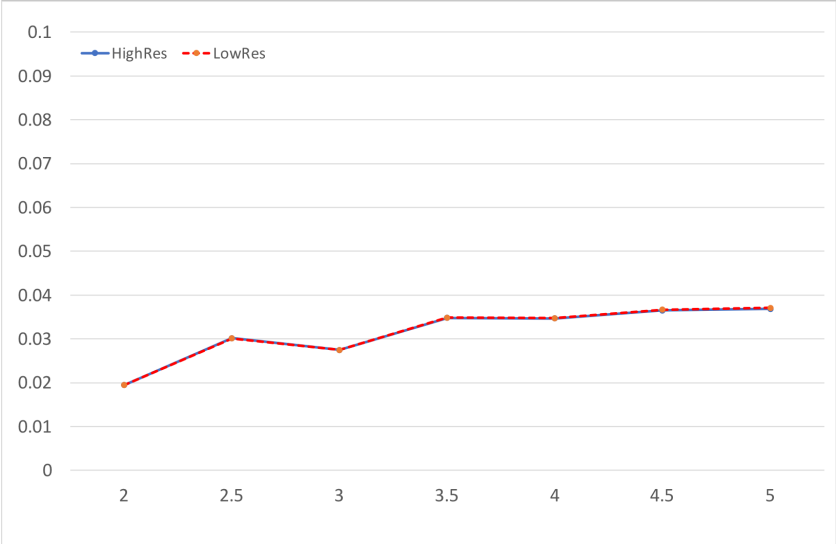


Figure 5.26: RMSE values for the indirect light from scene geometry. The X axis is the unit distances from the base position of the light source as it is moved towards the right side. Y axis is Normalised RMSE calculated from the entire geometry. The blue line for High-res geometry is almost invisible due to the closeness of values in both cases.

The Root Mean-Squared Error (RMSE) values in the graphs shown in Figure 5.26 has been calculated for the luminance at 8 different positions where the SLF were moved on the global X axis. The RMSE values are normalised by the range of the baseline values. As the light source has been moved gradually towards the right hand direction of the scene, the error values in overall luminance has been calculated for indirect light bouncing off the scene geometry. Unlike the previous evaluation, the graph does not include separate red, green and blue channels but only the overall luminance for low and the high resolution geometry.

The graph shown in Figure 5.26 shows the nRMSE values for high-res and low-res geometry in blue and a dotted red line respectively. Evidently the values are so close together, they are visually almost impossible to distinguish separately. This is why a separate graph at Figure 5.27 shows the improvement of the RMSE from ow resolution geometry to the high resolution geometry for UPoint distributions. The highest improved result is near the edge at $x = 5$ however, the improvement is only 0.65% which is negligible. It thus can be safely said that the high resolution

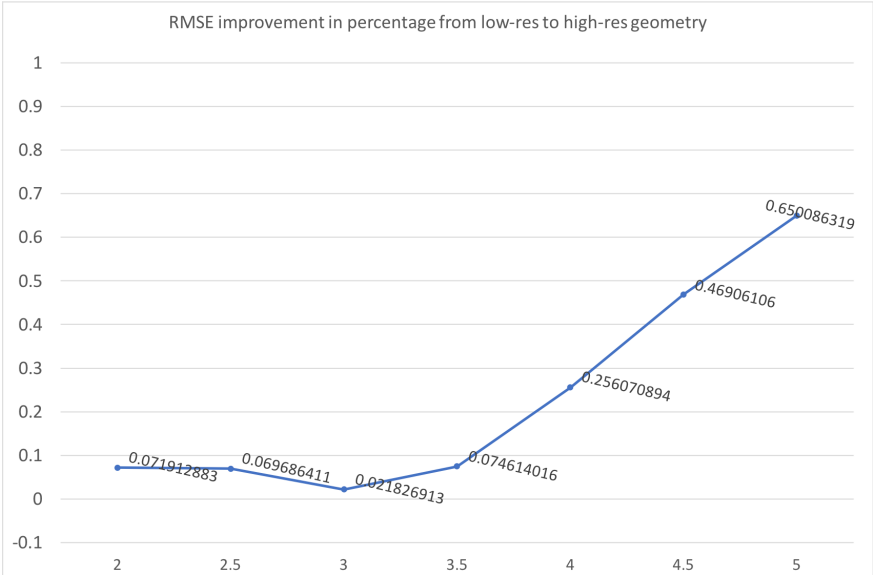


Figure 5.27: Improvement (in percentage) in RMSE values for the indirect light from scene geometry while using the high resolution geometry instead of lower resolution. The improvement percentage definitely increases as the lights are moved towards the boundaries but the amount is insignificant.

geometry acquisition do not have any significant improvement in DCP performance whatsoever. For some extreme cases that may affect DCP performance, the capture must be done in a way that circumvents those problems. Section 5.3.3 discusses this issue later.

The Absolute Percentage Error is the second type of error similar to the previous evaluation. It has been calculated throughout the space span of the scene geometry; i.e. the $((V_{original} - V_{moved})/V_{original})$ ratio for the different areas of the captured geometry. In the graph at Figure 5.28, the errors for the high-res and low-res geometry has been shown in blue and red dotted line similar to the RMSE graph.

The average absolute percentage error can be seen increasing in the graph described in Figure 5.28 as the light source is moved towards the edge of the scene geometry. Again, it is apparent that the error values for the two different geometry resolutions are extremely close together and there is absolutely no apparent trend in the improvement form the lower to the higher resolution geometry for the absolute percentage error. This further strengthens the conclusion that there is no significant improvement in accuracy for high resolution geometry data for simple scenes. Lower resolution geometry on the other hand can be acquired with significantly much more ease than acquiring precise geometry.

The previous evaluation explored the accuracy for change in the radiance of

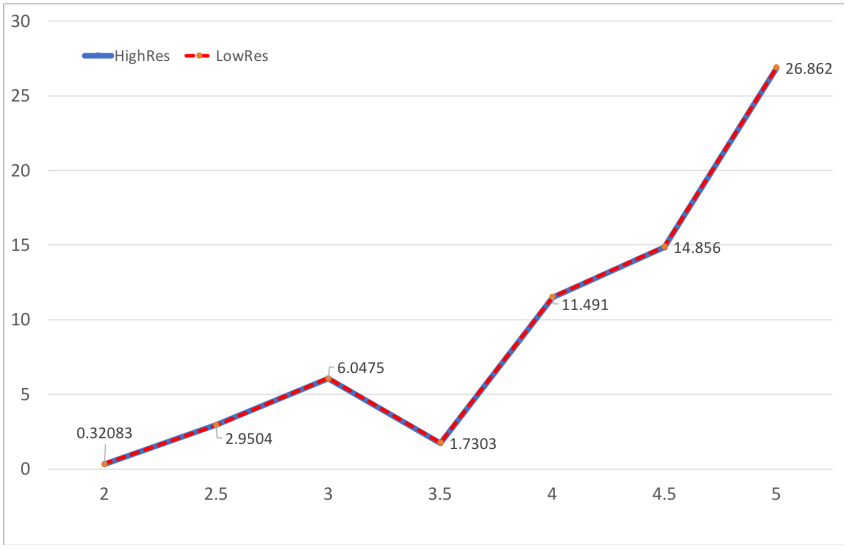


Figure 5.28: Absolute percentage error for movement of light source. The X axis is the unit distances from the base position as the light source is moved towards right side. The Y axis is the percentage of error calculated from the entire geometry.

the light source by comparing indirect light data with 9 different levels of change. The highest percentage of absolute error was 8% at 100 times of the base radiance. This is considerably accurate and thus there were no need to re-do the radiance evaluations for the high and low resolution geometry as well, particularly, given their extremely similar performances in the movement comparison.

Similar to the previous evaluation, this was conducted with a synthetic environment given the freedom and precision to control the position and radiance of the light sources.

5.3.3 Discussion

The objective of the DCP technique was to change the direct light in ILF with indirect light fidelity. The evaluations of DCP shows average absolute percentage errors of 9.14% and RMSE of 0.017 in the test cases when the light source is moving. These error percentages are below the JND which means, the indirect light difference coming from the scene geometry from an actual ILF and an edited ILF with the DCP will not be distinguishable by naked eye.

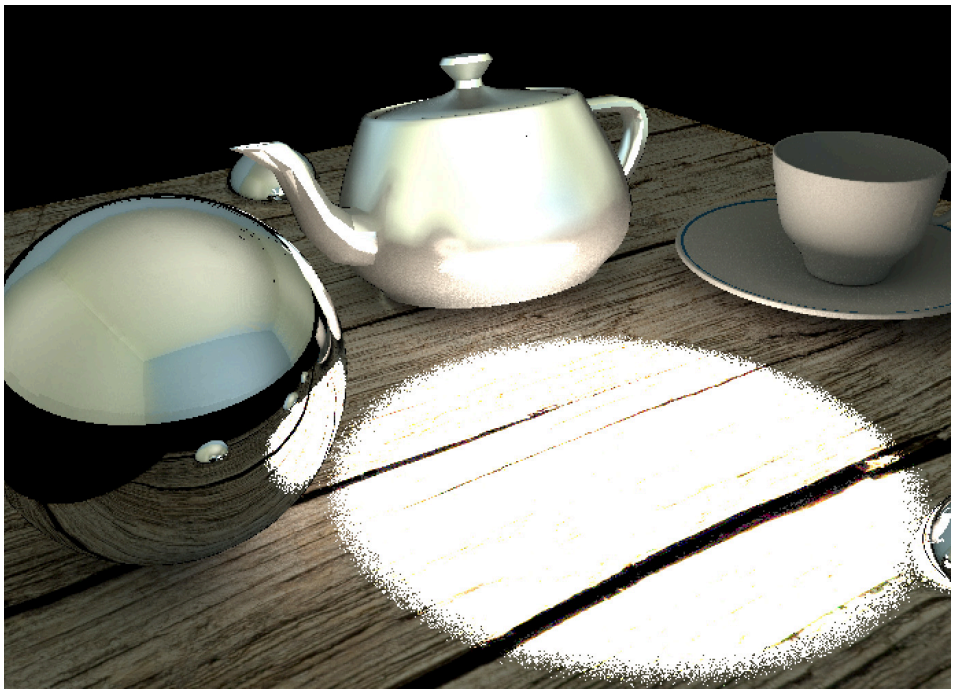
Figure 5.29 at the end of this chapter shows a few sample scenes rendered as an example of the DCP technique at work with changing light source positions and the intensities. The DCP technique propagates the dynamic changes made to the light source to the entire indirect light stored in the ILF database. The change in the indirect light with DCP can be seen in the synthetic mirrored balls reflecting the environment in these rendering samples.

This not only solves the problem for editing ILF, but subsequently opens up many new research directions as well in terms of object relighting and video relighting in post processing. Next section will conclude this chapter.

5.4 Conclusion

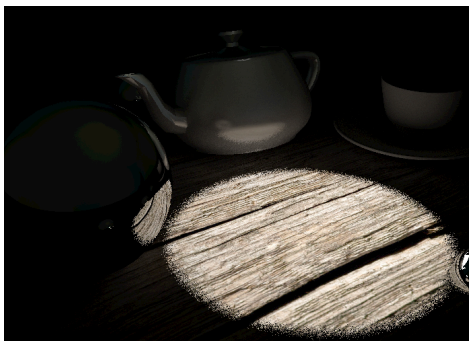
Incident Light Field is extremely useful for providing real-world lighting for high-fidelity rendering of synthetic scenes. It stores and renders with spatially variant lighting information which is crucial for photorealism in renderings. However, if a light source is changed in the scene, the entire indirect light of the scene also changes and thus the ILF needs to be recaptured. The way the ILF is captured is cumbersome and makes it impractical to re-capture it every time there is a change in the light sources. The DCP technique described in this paper enables the light sources in a pre-recorded ILF data to change not only the direct lighting but the indirect lighting as well by propagating the changes to all the other parts of the ILF data.

The DCP technique is a stepping stone towards solving one of the major limitations of the ILF, the lack of temporal capture. The next section will describe the Temporal ILF technique.



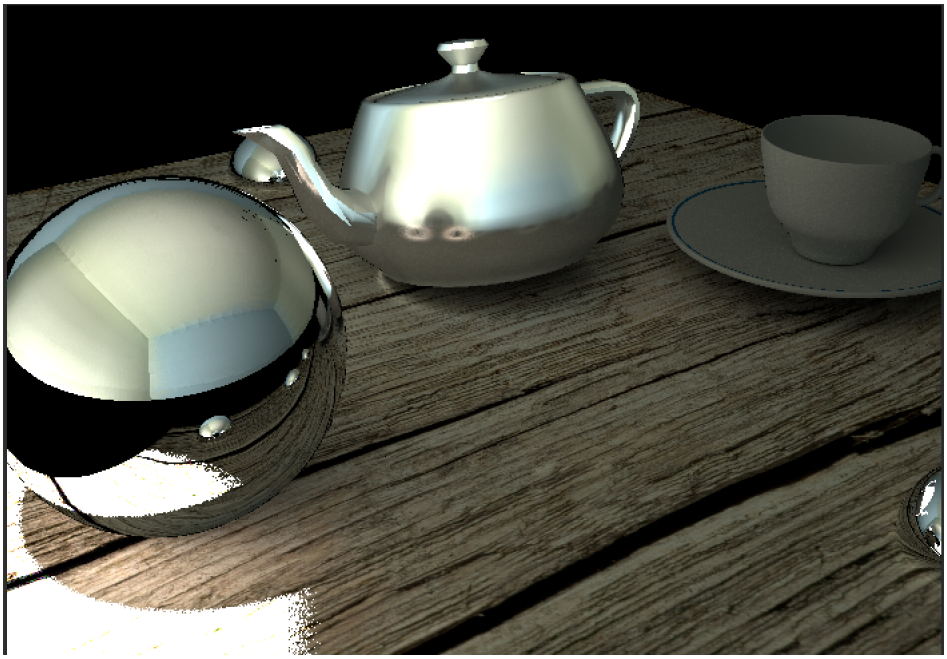
(a) Light power 100%

(b) Power 50%

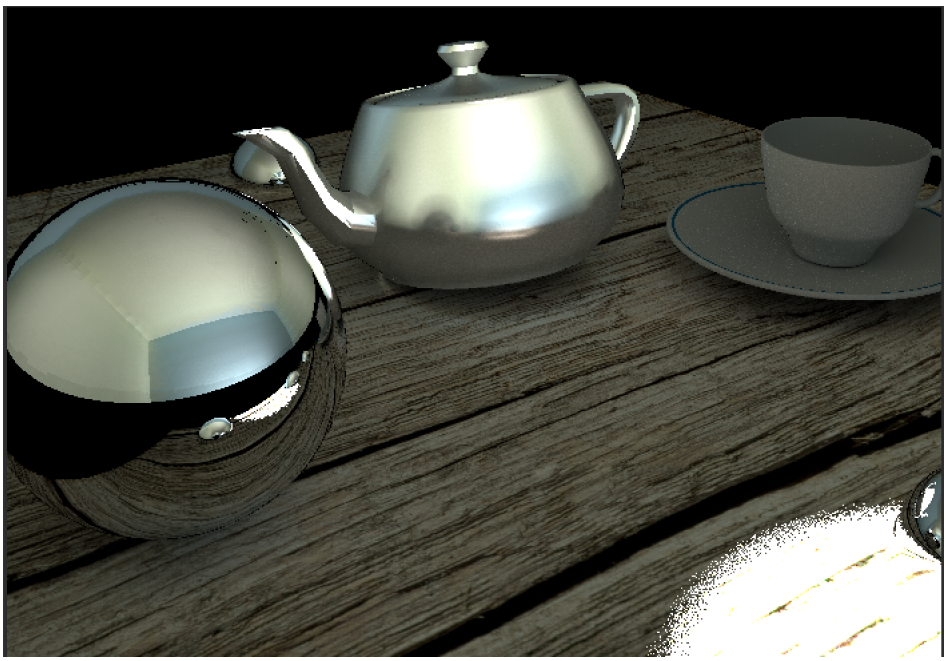


(c) Power 10%

Figure 5.29: Dynamic Change Propagation. A single ILF capture has been used to render the scene. The light source has been changed in power (a,b,c). The DCP technique propagates the changes the indirect light from the ILF capture. This can be especially seen prominently in the reflection on the mirrored sphere.



(a) Position 2



(b) Position 2

Figure 5.30: Dynamic Change Propagation contd. Same environment with changed spotlight position. Note the indirect light changes on the mirrored ball more clearly.

Chapter 6

Temporal Incident Light Fields

"I may not have gone where I intended to go, but I think I have ended up where I needed to be."

-Douglas Adams

Figure 6.1: Temporal snapshots of a synthetic object rendered in a dynamically changing spatio-temporally varying light. Temporal ILF has captured and rendered the synthetic scene with the spatially varying spotlight which is dimmed over time. It is noteworthy that the indirect lighting form the environment also changes accordingly.

The Temporal Incident Light Field (Temporal ILF) is a novel technique which captures and represents spatially and temporally variant light. Previously, Chapter 3 provided an overview of the developmental flow (Figure 3.1) from the ILF technique described in Chapter 4 to the DCP technique described in Chapter 5 and finally to this chapter describing the Temporal ILF which uses both ILF and DCP to capture and render synthetic objects with complex real-world light changing dynamically over time. The next sections will describe the general working principle of the Temporal ILF. The pipeline has three steps: Capture, Representation and Rendering. Section 6.2 describes the Capture and Representation while Section 6.3 describes rendering with Temporal ILF. This chapter also includes a real-world implementation of the Temporal ILF in Section 6.4. Finally, Section 6.5 states the current limitations and draws the conclusion.

6.1 Working Principle

Real-world light is dynamic and can change across space and time. Capturing spatio-temporally varying light over large areas and rendering synthetic objects with it is the primary objective of this work. The existing ILF technique was the first step towards this direction, having abilities to render spatially varying complex real-world lighting. The ILF captures however cannot be used as instantaneous snapshots over time to represent temporality as the capture process does not have a single point of operation but spread across the capture space. Its inability to capture instantaneously makes it infeasible for using as temporal snapshots of the spatial variations over time.

The novel Temporal ILF technique described here circumvents the infeasible requirement for temporal ILF snapshots by synthetically editing the light sources in a single ILF capture rather than capturing the ILF of a scene over and over each time there is a change in the light sources. The core working principle captures an initial ILF once, records the positional and intensity changes in direct light sources and edits the initial ILF synthetically via the DCP technique with indirect light fidelity to produce any arbitrary temporal snapshot of the ILF at a certain point of time. Chapter 3 has introduced the Temporal ILF briefly along with the interconnections of the ILF and the DCP technique earlier. This section will provide a detailed description of the Temporal ILF.

6.1.1 Applications

One of the primary applications of Temporal ILF will be the CGI rendering in *motion pictures*. Currently the state-of-the-art in movie industry does not usually render synthetic objects with real-world spatially varying lighting and assumes a uniformly distributed lighting instead. Temporal ILF would enable the motion picture industry to use and embrace complex, real-world light.

In addition to the motion pictures, Temporal ILF would be very useful in virtual reality (VR) and augmented reality (AR) applications. Normally the synthetic objects that are rendered in the real environment are small enough to assume uniform light distribution along its surface. Accurately rendering large objects in a spatially varying light changing over time is currently infeasible and usually avoided in commercial applications. The Temporal ILF technique will be able to render objects of any size into a dynamically changing environment. The next sections will describe the methodology of the Temporal ILF, and its real-world implementations.

6.1.2 The Temporal ILF Pipeline

The Temporal ILF has three parts: (i) Capture, (ii) Representation, and (iii) Rendering. Figure 6.2 provides a schematic diagram of the Pipeline. The first step is

Capturing the Temporal ILF. This has three separate parts of its own: (a) capturing the scene geometry where a crude geometry of the environment is acquired, (b) capturing a static ILF where the all the direct light sources are switched on, and lastly (c) tracking the position and the radiometric properties of the light sources over time.

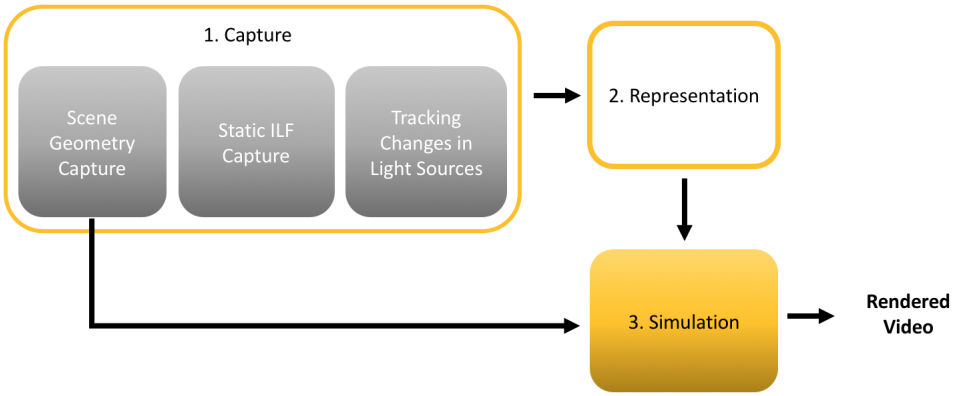


Figure 6.2: Temporal ILF pipeline.

The second step, Representation, processes of the data captured in the first step of the pipeline and represents them in their respective formats. Firstly, the captured ILF and the tracking data is processed and represented into subsequent formats. Details about the ILF representation and ILF ray-database construction can be found in the Chapter 4. The acquired geometry however aids the DCP implementation in part three directly and does not need any additional processing. Section 6.2 described the Capture methods and the representation of captured data in proper formats.

The third and final step is the Rendering. In this step, the light sources in the static ILF are changed synthetically according to the tracked changes with the help of the DCP technique described in Chapter 5. DCP helps changing the light sources as well as updates the effects on the indirect lighting in the ILF. The result is a modified ILF of the scene with the changed lighting environment as recorded at any given point of time. This way, given a time period, a rendered video can be obtained where each frame in it has been illuminated by a different ILF with different lighting conditions. Section 6.3.2 describes the rendering with the Temporal ILF pipeline in detail.

6.2 Capture and Representation

As described above, Temporal ILF captures the scene geometry, the static ILF and tracks the light sources over time.

6.2.1 Capturing Geometry

The scene geometry is a crucial part of the DCP technique described in Chapter 5. DCP estimates the indirect light from any arbitrary position of the direct light sources with the help of the scene geometry. Section 5.2.1.1 describes the geometry acquisition technique used in DCP. As mentioned there, acquiring a scene can be an expensive and complex process depending on the complexity of the scene and the target accuracy. There are laser based point cloud generator devices available such as the FARO Focus[1], which provide a very high level of detail in scene geometry. This high level of detail is useful for many other applications but the DCP technique does not require extremely precise geometry where every object, nooks, troughs and ridges in the scene geometry is accurately captured. The evaluations of DCP shows that the estimation of the indirect lights form the light sources do not change above the Just Noticeable Difference (JND) in case a crude scene boundary is used in place of accurate scene geometry information. In validation, Section 5.3.2 describes an evaluation where a scene with ridges and troughs were considered for the change propagation. There were no significant improvement (the RMSE improved by 0.65% in the test case) in the accuracy while using the accurate high resolution geometry from a crude boundary of the scene. The scene geometry capture in Temporal ILF is only to aid the DCP in the final step, *simulation*; thus the scene boundaries are captured in a simpler way than expensive equipments such as laser scanning approaches.

Scene boundaries can be acquired in any way which is simple yet effectively accurate enough for the present work. Uses of IR based tracking sensors such as the Microsoft Kinect can be a viable and affordable option, subject to availability[2]. The implementation of Temporal ILF in this work however uses stereo image pairs and a laser distance scale to get the real-world scene boundaries. Figure 6.3 shows the distance meter and a mounted stereo rig which holds two cameras. The distance between the cameras can be changed with the rig. For these experiments the distance between two cameras were $19cms$ while the images were $25cm$ apart. The boundaries are calculated using the process described previously in the Section 5.2.1.1 as this also aids the DCP technique. The distance meters were used additionally as the

[1] `https://www.faro.com/en-gb/products/construction-bim-cim/faro-focus/?gclid=CjwKCAjwypjVBRANEiwAJAxlIs1xKyHhU2JcC6KPzWJmNEcZ9GDJJNEp4XxMGs5K70wJtGFcMfUG3hoCPmkQAvD_BwE`

[2] `https://www.theverge.com/2017/10/25/16542870/microsoft-kinect-dead-stop-manufacturing`

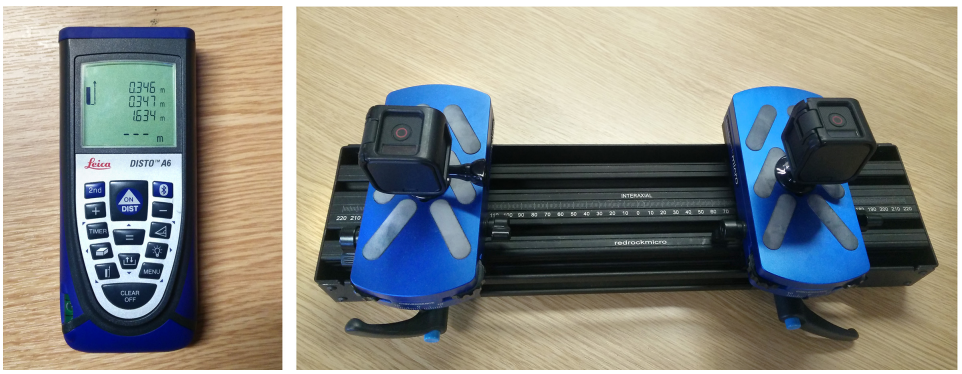


Figure 6.3: The Leica laser based distance meter and to cameras mounted on the stereo rig.

initial known-distance-calculator and later a redundancy to verify the calculated dimensions at different parts of the scene acquired by the stereo image pair setup. For simplicity, this redundancy can be readily skipped depending on user preferences, or any other method of choice can be implemented seamlessly.

6.2.2 Capturing Static ILF

The static ILF capture at the beginning of the Temporal ILF capturing process acts as the foundation to this technique. As capturing actual ILFs at each point of time is unfeasible, Temporal ILF uses synthetically edited ILFs as snapshots throughout the capturing time. The ILF captured in the beginning of Temporal ILF capture and act as the basis for all the synthetic changes that are going to be made in it during rendering with the Temporal ILF.

The general capturing process is exactly similar to the regular ILF capture as described in Chapter 4, apart from a small difference in the sampling area. Originally the ILF capturing can be done only along the rendering space, i.e. only the part where the synthetic object will be rendered. Even in a big area with multiple light sources, traditionally static ILFs could be captured only in the area of interest. Figure 6.4 explains the situation with the schematic diagram of a sample scene. There are three light sources L_1, L_2, and L_3. However, as the rendering space or the area of interest marked by A is located at one side of the entire room, the ILF is only captured within the area A.

In contrast to this, a major strategical difference for Temporal ILF is the requirement of covering all the light sources in the environment in the base ILF capture as any one may change arbitrarily. Even in cases where the distant light sources are not changing, the base ILF must cover all the direct lights to enable DCP technique to estimate the indirect light in the scene. It is imperative that during this capture all the lights are turned on. Any light source added to the scene after

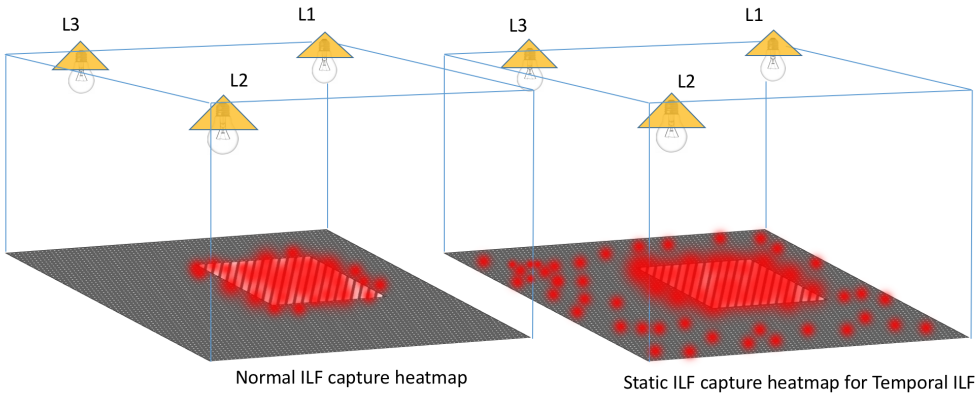


Figure 6.4: A heat-map visualisation of spatial sampling for normal ILF versus static ILF for the Temporal ILF.

the base ILF capture will not have its effects reconstructed in the Temporal ILF. At the scene shown in Figure 6.4 the light L_3 is outside the rendering space however the base ILF will have to capture the ILF in the whole scene. Fortunately this capturing area augmentation comes with no extra resource requirement and negligible extra effort. The free form ILF capture [Unger et al., 2008] has shown that the capture can be spread across any arbitrary volume in space with arbitrary density; i.e. the light probe density can be very low outside the rendering space. Figure 6.4 shows the sampling density of both normal ILF and and the static ILF for Temporal ILF capture schematically as a heat-map.

6.2.3 Tracking light sources

Tracking the changes in the light sources is an unique addition to the Temporal ILF capture. The direct light sources are tracked for the positional and the radiometric changes (intensity and colour) against elapsed time. There are mainly two approaches of tracking the light sources that this work proposes: (i) light probe approach, (ii) marker detection approach. Due to the unavailability of expensive HDR video cameras, this work only implements the second approach and only proposes the former.

6.2.3.1 The light probe approach

This approach tracks lights via an unidirectional light probe video. One or more HDR video light probes can be used for this method on the rendering surface directly

looking at the light sources. The light probes can be the traditional mirrored ball [Debevec et al., 2000] or an HDR video camera with a fisheye lens. Figure 4 shows a schematic diagram of this method. High energy regions in the captured tracking video can be easily tracked via several existing algorithms or ready-made Computer Vision based libraries.

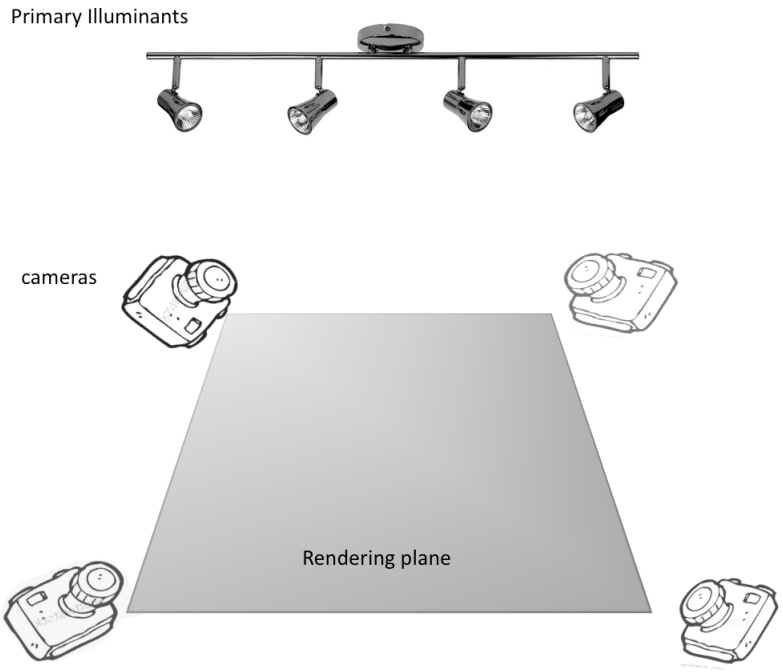


Figure 6.5: Tracking changes with the light probe approach. TODO This bad diagram will be changed.

The main advantage of tracking light sources with this method is it can track the position of the lights as well as the dynamically changing colour and intensity of them. However, a major drawback of this approach is the equipment and the setup. HDR video cameras are not yet common and the consumer-grade HDR video cameras such as ARRI Alexa are very expensive. Moreover, these cameras are not portable enough to set up easily in an environment. Thus, acquiring several such cameras are financially and logistically challenging. The second approach attempts to eliminate this problem.

6.2.3.2 The marker detection method

This method is a much less expensive approach to track the light sources than the former and does not need HDR video cameras. Each of the light sources are fitted with different ArUco marker images [La Delfa et al., 2015]. ArUco [Garrido-Jurado

et al., 2014] is a highly reliable fiducial marker detection system which can be easily detected by existing computer vision and pattern recognition algorithms with high level of accuracy. This work has used exactly this marker based detection technique for the light probe position detection while capturing ILF described in Chapter 4. These markers can be printed in a paper and can be attached to any object very easily. Barring some extremely dynamic circumstances, it can be predetermined which light sources are fixed to its position and will not change their position over time. Markers are not attached to these sources, nor are they tracked unnecessarily. Inexpensive LDR video cameras can be used to track these markers over time.

The major drawback of this is, currently the marker detection approach can only be used for detecting the positional changes accurately in the light source. It cannot determine the amount of radiometric changes automatically and only registers the intensity changes as *high* or *low* in the processed data. As in the real-world implementation of this work it was not possible to acquire multiple expensive HDR cameras, the current setup records the radiometric changes manually by a light-meter and predetermined settings in the light sources while dimming or switching off a particular light source. This is one of the current limitaiton of the implemented Temporal ILF technique. There is an ongoing work in progress to extend this approach to detect the radiometric changes with an acceptable level of accuracy automatically. Section 7 discusses the plans for the future.

6.2.4 Processing

The raw data from the captured ILF and the tracked light sources need to be processed to obtain the appropriate representation ready to be used for the renderer.

6.2.4.1 Processing ILF light probes

The initial base ILF capture produces sets of HDR light probe images which are processed similar to the method described in Chapter 4. As there were no video HDR camera available for this work, bracketed still LDR images were taken with a fisheye lens mounted on a digital camera which were then merged into HDR images. The marker based detection described in Chapter 4 has been used again to detect the global position of the captured light probes in the environment. The image and position pairs are converted next into the ILF ray-database using the method described in Chapter 4.

As described previously in Chapter 4, the ILF ray-database is constructed with reference to the "ILF planes". The dimensions of these ILF planes originally only needed to form a convex hull around the rendering space and they did not need to adhere to the actual scene boundary dimensions. However, as Temporal ILF uses the DCP technique, the base ILF needs to be mapped to the scene geometry. The

ILF planes in this case are constructed according to the scene boundaries acquired in the Temporal ILF capture. This way, there is an easy and direct mapping to the ILF data and the corresponding scene geometry. The illuminance data form the light sources are separately stored in Source Light Fields (SLF) as described in the Chapter 4 and mentioned later in the Chapter 5. The position and dimensions of these SLFs are also constructed according to the real-world position and dimensions of the corresponding light sources. The result is a base ILF ray-database that is editable by the DCP technique and ready to be used by a physically based renderer.

6.2.4.2 Processing of the Tracking Data

The Temporal ILF capture tracks the position and the intensity of the light sources that are changing in the scene in the given time period. The primary objective is to obtain the position and intensity of all the light sources at any given point of time. As mentioned earlier, there are two approaches to track the light sources. The first one is a proposed method with high accuracy, however expensive equipment has restricted its implementation in real-world implementations done in this PhD. The second method is the marker detection method where the position detection is automatic and but the intensity and colour detections are manual.

Both the tracking approaches acquire the position and the radiometric properties of the light sources in each frames of the recorded video. Each of the frames are timestamped to avoid confusion. During the position detection, a custom software detects the position of each light sources and produces the data in the format shown in the diagram in Figure 6.6. For example, for the first frame, there are two pieces of data for the timestamp and the number of light sources (SLF in ray-database) detected in the frame. Each such source has three data entries for the "ID" of the light source as well as the colour and the position. The IDs are integers (ideally starting from zero) mapped to corresponding light sources. The cardinality of the detected light sources per frame may not be the same as it only records a light source if there is a change detected in it.

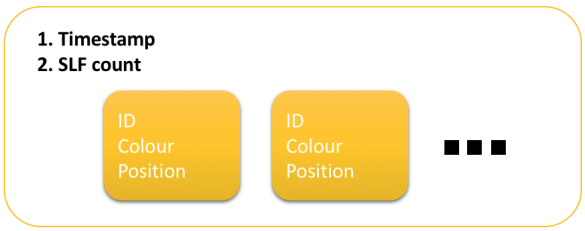


Figure 6.6: Tracking data format.

A custom software is used offline to extract the tracking information from

an input video. Each frame of the video produces the the tracking data in the mentioned format. A stream of such tracking data connected by the timestamps is obtained from the video which then is used by the renderer in the simulation part of the Temporal ILF pipeline.

6.3 Rendering

The objective of Temporal ILF is to produce a rendered video of the duration of the temporal ILF capture. The final part of the Temporal ILF pipeline is rendering where the changing light properties over time is used to simulate the change the position, intensity or colour of the concerned light sources synthetically in the base ILF. Each frame of the output video is rendered by a *snapshot* of the base ILF with the changed light sources. The tracking data includes the change information for all the frames in the intended video. An output video could be rendered with the following algorithm:

Algorithm 5 Temporal ILF Pipeline: Simulation

```
1: procedure VideoRenderer(ILFhandle, Scene, TrackingData)
2:     td ← read(TrackingData);
3:     i ← 0;
4:     while i < td.TotalFrames do                          ▷ reading i^th frame
5:         j ← 0;
6:         while j < td[i].SLFcount do                ▷ from j^th SLF of i^th frame
7:             updateSLFdatabase(td[i].SLF[j].ID, td[i].SLF[j].Colour,
8: td[i].SLF[j].Position);
9:             j++;
10:        end while
11:        updatedILF ← DCP.propagateChanges(ILFhandle, Scene);
12:        renderFrame(updatedILF, Scene);
13:        saveFrame(td[i].TimeStamp);
14:        i++;
15:    end while
16: end procedure
```

Algorithm 5 describes the procedure *VideoRenderer* which renders a single frame of rendered image by iterating through all of the frames of the tracking data. for each $frame$ in the tracking data, it again iterates through all the SLF (light sources) recorded in that frame and updates each of these SLFs with the recorded colour and position by their IDs. After all of them are updated, the VideoRenderer calls the DCP technique to propagate the changes just made to the SLF to the

indirect light stored in the ILF database. the *DCP.propagateChanges* procedure produces the $updatedILF$ which is then used to render the scene for the frame and the image is subsequently saved to the disk with the Timestamp for the current $frame$.

6.3.1 DCP Implementation

The DCP technique has been described thoroughly in the previous chapters. It has primarily been designed for the Temporal ILF in order to dynamically change the light sources synthetically. It however has been developed in a modular fashion to facilitate flexibility in the implementation. The *Temporal ILF engine*, the *ILF engine*, and the *DCP engine* are three interconnected modules.

The DCP engine takes the change informations from the tracking data stored in temporal ILF engine. It then interacts with the scene geometry and changes the ILF ray-database accordingly via the ILF rendering engine. The schematic diagram at Figure 6.7 depicts the entire process of rendering videos including the interworking and information flow inside the Temporal ILF engine.

6.3.2 The Temporal ILF Engine

The Temporal ILF engine encapsulates the tracking informations along with the ILF engine and the DCP engine. The ILF engine has an ILF interface, the ILF slab geometry (which is also the Scene geometry in this implementation), and the ILF database. The interface can interact with any arbitrary renderer and output colour value from the ray-database for a given ray origin and direction. The renderer can query the Temporal ILF engine for each frame and produce a rendered HDR video of the synthetic scene with real-world spatially and temporally varying light.

The entire end-to-end process has been described in the Figure 6.7. The green arrows are the data flow in the outermost layer, passing inputs to the renderer and getting the output video. The video is produced as individual frames. There is a frame counter as a part of the renderer which triggers the render of each frame after the completion of the previous for the entire duration of the output video. For each such frame, it interacts with the Temporal ILF engine which in turn extracts the relevant tracking data for that particular frame and passes the information to the DCP engine for processing. The DCP engine interacts with the scene geometry and the ILF ray-database to change the position, colour or intensity of the light sources as mentioned in the tracking data and updates the ILF ray database accordingly. This part of the data flow has been depicted in the diagram with black arrows.

As the ILF is updated for the current frame, the renderer then triggers the ray-tracing engine (or any variant) and produces a ray which interacts with the synthetic scene to produce a *hitpoint* and a ray "direction". This $(hitpoint, direction)$

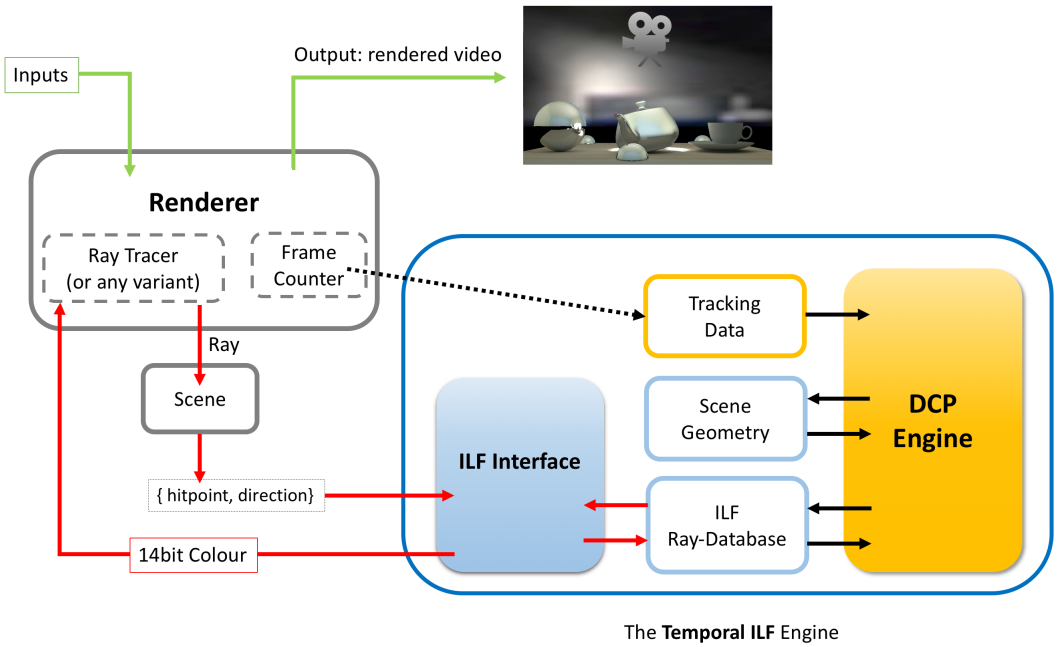


Figure 6.7: End-to-end rendering with Temporal ILF engine.

pair is then passed to the Temporal ILF engine which interacts with the ILF interface within it to calculate the appropriate colour value for the given hitpoint and the ray direction. This part of the data flow has been depicted in the diagram with red arrows.

The result is a frame by frame rendered video of the synthetic scene with a time variant, complex real-world lighting. This is shown in the diagram by the rightward facing green arrow. The next section will describe a real-world implementation of the Temporal ILF.

6.4 Real-World Implementation

The Temporal ILF has been implemented with a real world scene with spatially varying lighting that changes over time in terms of intensity and position. Although as mentioned before in Chapter 5, the limitations in the available equipment were prohibitive of acquiring the base ILF with adequate light probe samples, this implementation uses the static HDR light probe method to capture 1173 light probes across the test surface. The end results were not comprehensively same as the actual scene due to low sampling density, however, the light boundaries clearly adheres to the original real-world scene.

6.4.1 Capture

The capture has been done in a rectangular room (see Figure 6.8) with two fluorescent area light sources and a spotlight. Figure 6.8 shows a schematic wireframe of the room. The area lights are fixed in position but they can change the intensity to three different levels, i.e. 100%, 50%, and 25%. The spotlight can move its position.

6.4.1.1 Capturing the Scene Geometry

The scene geometry has been captured similarly to the processes described earlier in Section 6.2.1. A distance meter is used in tandem with the stereo capture rig. The room consisted of several objects such as chairs along its boundaries however, it was established in Chapter 5 that the minor alterations such as those do not affect the result of the DCP technique. The geometry capture process revealed the room with the following dimensions: Height 3.76 meters, Width 5.23 meters with the Length 7.44 meters.

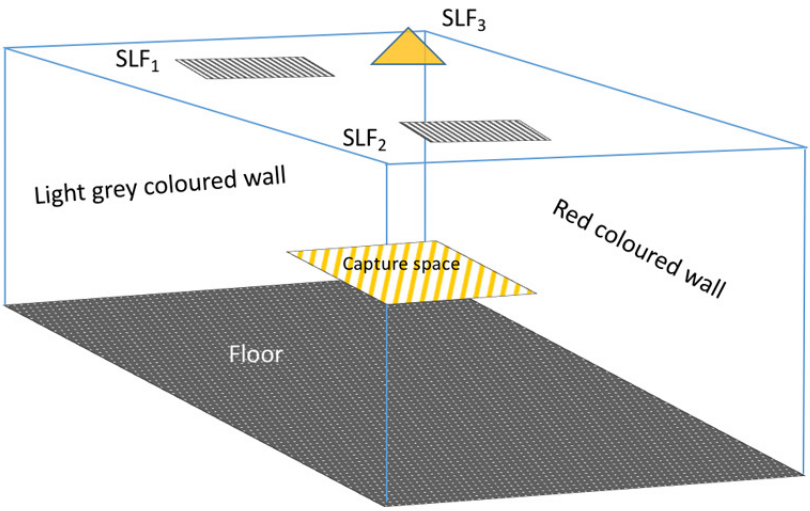
6.4.1.2 Capturing the ILF

The ILF has been captured on a surface of a large table. As there were no HDR video cameras available, the ILF was recorded by manually moving a Digital SLR camera on a slider with a fisheye lens and capturing 7 bracketed frames which were later merged to a single HDR file. 1173 of such HDR light probes had been taken throughout the capture space for one ILF capture.

The capture follows the process described in Section 6.2.2. Figure 6.9 shows the setup where the camera with the fisheye lens is facing upwards attached to a slider. This manual ILF capture took around 7 hours of capturing. The total number of probes were 1173. This is about 50 times less than the 50,000 probe capture that Unger et al. did with their specialised capture rig around a similar dimension surface [Unger, 2009]. This amount of manual effort is a hindrance for capturing multiple complex scenes, however access to proper resources such as a good quality HDR video camera with a reasonable frame rate will alleviate the time and effort that was spent for this capture in future.

6.4.1.3 Tracking the Light Sources

Section 6.2.3 already proposed two possible methods for tracking the light sources, their changes in position, colour and intensity. The *light probe approach* could not be implemented due to the limited resource available. The *marker detection method* has been implemented with an inexpensive digital video camera recording the movements of the light sources. A printed ArUco marker [Garrido-Jurado et al., 2014] has been



A wireframe schema of the real scene

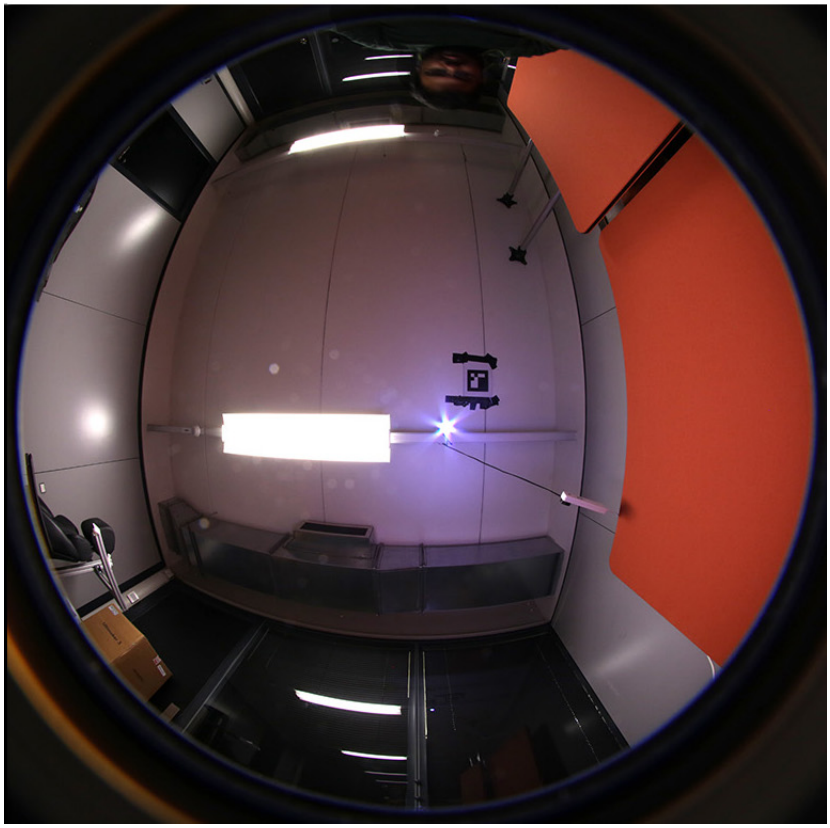


Figure 6.8: The scene used for the Temporal ILF capture. The top is the wireframe of the scene. The below is a light probe from the scene. The SLF_1 and SLF_2 are the two fluorescent area lights. SLF_3 is a spotlight. The capture area is a table elevated 1.9 meters from the Floor. The right side wall has two diffuse red painted screens.

attached to the spotlight for easy detection and accuracy. The fluorescent lights were not fitted with the markers because they did not change their position.

Figure 6.9: The camera setup for ILF capture.

6.4.2 Rendering

The processing and the rendering methodology has been described in the Section 6.2.4.2 and Section 6.3 respectively. For the rendering, this work used an in-house path-tracing renderer developed for this work that interacts with the Temporal ILF engine. The simple synthetic objects used are mostly designed in house. The complex object models (the cup and the Venus) were obtained from the Stanford university 3D model repository. The result images in the end are part of the tracked change sequence in the original scene.

Figure 6.10 shows a render of the synthetic scene with wooden textured base. all lights are switched on. Figure 6.11 and Figure 6.12 shows various positions of the spotlights moved around synthetically. The first frame in Figure 6.13 shows render from only spotlight switched on and area lights turned off synthetically. The second frame shows a colour bleeding from the right side wall which was painted red. The spotlight was synthetically focused on that wall and the Temporal ILF engine updated the indirect light accordingly to have red coloured indirect light bleeding form the wall. The mirrored balls in the back also shows the automatically edited right side wall as the result of the spotlight being focused there.

6.4.3 Implementational Limitations

1. Inadequate sampling density of base ILF. The spatial variation of the light is recorded but failed to produce the crisp boundaries as in the original scene.

2. Lack of proper lighting equipment and control over them made some of the features such as colour bleeding while moving the light around has been restricted to certain extent. The spotlight needed to comletely focus closely on the right

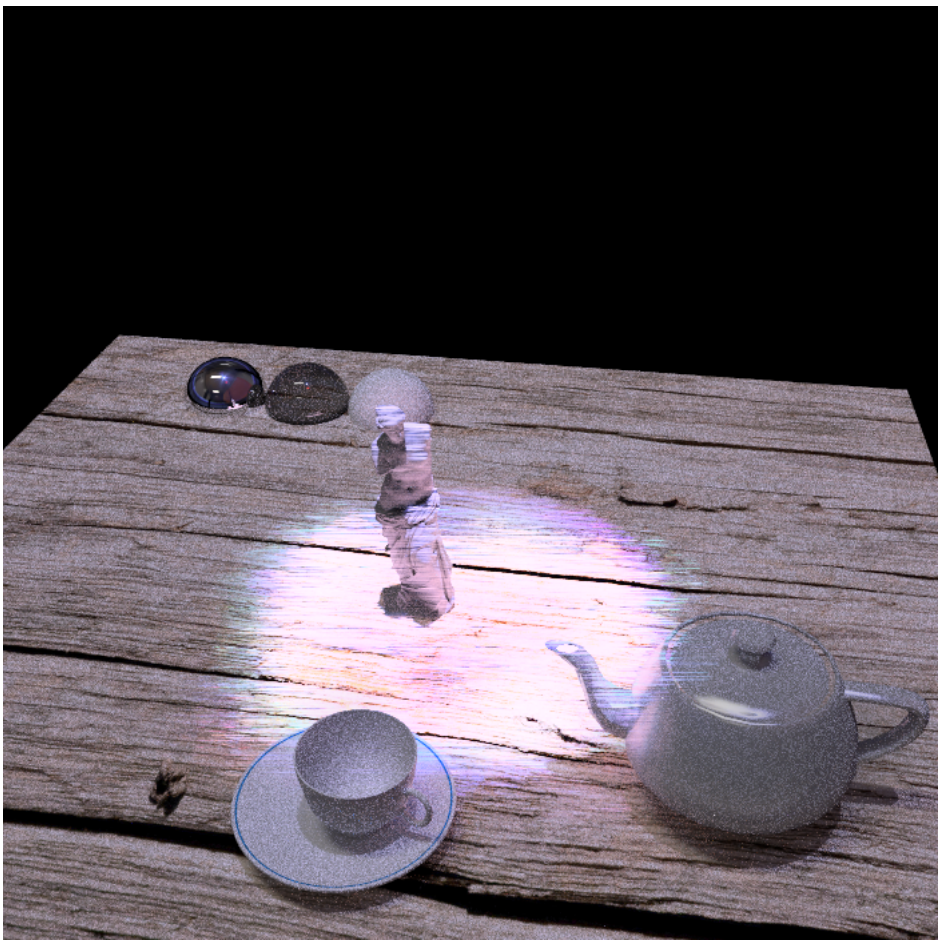


Figure 6.10: An Temporal ILF render of synthetic scene with real world light. Relative position of light is (0,0,0).

side red painted wall to induce discernible colour bleeding as the spotlight used was a low powered one.

3. Extremely slow manual capture made it impossible to capture dynamic natural light like strips of sunlight which would have been aesthetically pleasing.

The following figures show the rendered images from the real scene. The next section will discuss the limitations and the future directions of the Temporal ILF.

6.5 Discussion

This chapter described the entire workflow of the novel technique "Temporal ILF" and presented a real-world testing of it. The next sections will describe the limitations and the future directions of Temporal ILF.

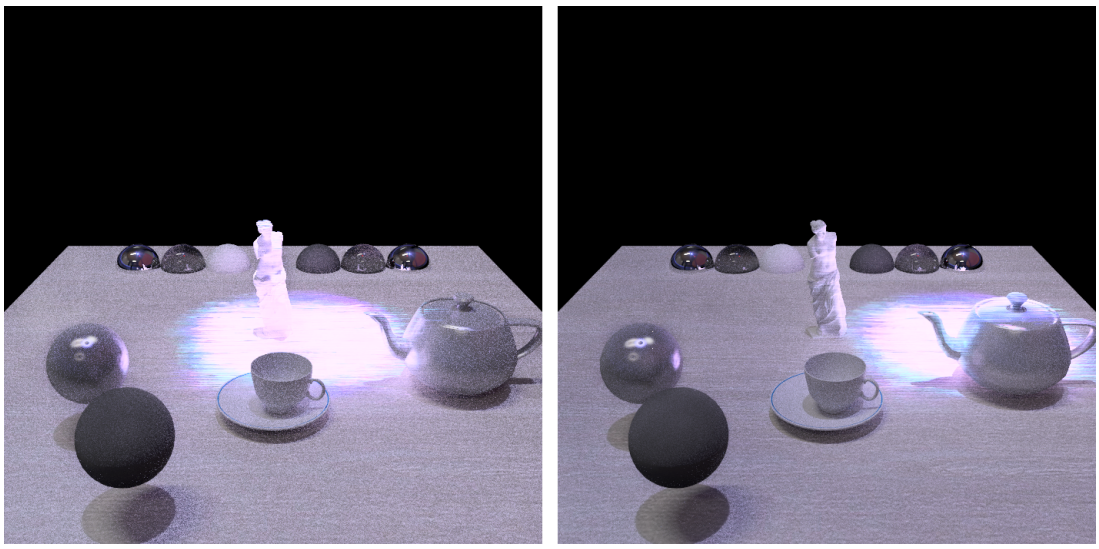


Figure 6.11: Temporal ILF renders with SLF movement (plain texture base). Relative position of light is left image (0,0,0), right image moved to (2.495,0,0.25).

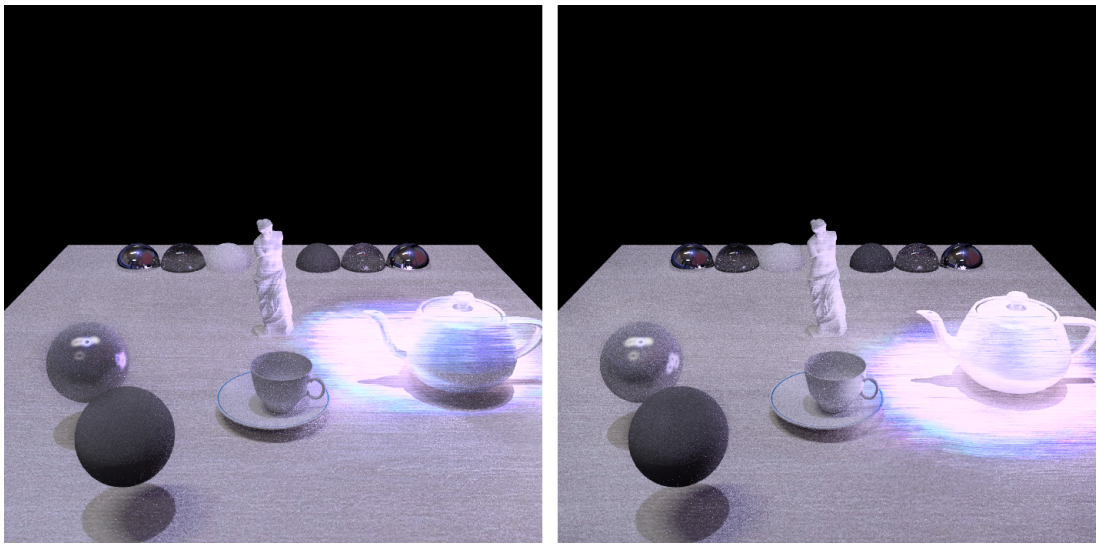


Figure 6.12: Temporal ILF renders SLF movement contd. Relative position of light is left image moved to (2.49,0,1.45), right image moved to (2.49,0,2.145).

6.5.1 Limitations

Temporal ILF currently is the only method for capturing and rendering with real-world spatially and temporally varying complex lighting. Although the results from the implementations are promising, there are a few limitations that could be addressed in the future.

1. **Capturing ILF:** The ILF capture is not a straightforward and easy process. It requires precision instruments and high quality HDR cameras which are very expensive. This has been discussed in Chapter 4 briefly and despite having

other alternative methods to capture ILF, most work in a small capture space. Capturing ILF in a arbitrarily large area in outdoors with the present methods is infeasible.

Moreover, determining the exact position of the captured light probe is a challenging task especially with free form ILF. The present methods described in Chapter 4 work best in spacial cases and are non-trivial for a user trying to capture ILF in a real-world scene because of lack of portability of the capture equipments, not to mention the financial constraints.

Temporal ILF suffers from the same problems with the static ILF. There is a huge scope of work to be done to make ILF captures portable and easy to implement.

2. **Scalability:** ILF originally was designed mainly for small scale captures and the capturing techniques were also designed in the relevant way. A few approaches make ILF capture possible for capturing large spaces, albeit with significant effort and complexity. The representation of the ILF is however not fit for handling very large captures. Even with ILF captures of reasonably small indoor areas, the size of the ray database can go up to tens of gigabytes depending on the spatial and angular resolution. This also has been briefly discussed in Chapter 4. For large scale captures, the huge memory requirement for the ILF ray-database makes it very hard for scalability.

   Temporal ILF suffers form this mainly because it is dependent on a dense ILF capture which it needs to change for every frame that is rendered in the resultant video. There is a definite scope of further research to compress the ILF data to make the ray-database more manageable.

3. **Tracking Lights:** The final limitation of Temporal ILF is the methods of tracking light sources. The two proposed methods of light source tracking have their own limitations. The *light probe method* requires HDR video cameras with a workable frame rate (24FPS preferable). The present work could not implement this method because of the resource constraints. The second, *marker detection method* is much easier to implement but lacks the ability to simultaneously record the intensity and colour changes accurately because of the limited dynamic range of inexpensive LDR cameras.

   A future research could be done to address this limitations which will make Temporal ILF much more portable and accessible to the general consumers.

6.5.2 Future Directions

The section above pointed out the three main limitations of the Temporal ILF. Solving each of these issues will require in-depth research and development on the capture, compression and tracking lights.

6.5.2.1 A Portable ILF Capture Device

A future research could be done on developing a portable device which can capture HDR light probes with a reasonable resolution and frame rate. Such a capture device should also be able to detect the global position of each light probe recorded.

For capturing large areas, different means of capture settings can be developed and evaluated for accuracy. For large outdoors, there are commercially available highly successful drones and quad-copters which are affordable and have reasonable positioning system based on state-of-the-art computer vision techniques.

Together with a ILF capture device and the movement technique, a future work can develop a scalable Temporal ILF framework that can capture large race-tracks or exotic locations for shooting commercial films or archeological spots in various parts of the world.

6.5.2.2 An Intelligent Tracker

The previous section described the main issues with the current light tracking methods. A potential work could develop an intelligent tracker system that not only eliminates the use of fiducial markers like the ArUco markers [Garrido-Jurado et al., 2014] used in this work. Moreover, the change in light can be tracked and measured even by inexpensive LDR camera sensors with some intelligent algorithmic solution that controls the exposure according to the changes in the lights so even a high intensity light source will not be "highlight-clipped" in the tracker camera capture.

6.5.3 Summary

- Temporal ILF is a novel technique for capturing and rendering with spatially and temporally variable real-world lighting.

- It uses a novel technique to track the changes in the light source over time with a video recording of the changing lights. This video is then processed and represented as a timestamped tracking data for all the available direct light sources in the scene.

- Temporal ILF engine uses the static ILF technique described in chapter 4 and the DCP technique described in Chapter 5 to change direct light sources in the

static ILF synthetically in a way so that the indirect lights are also affected by the change accordingly.

- The Temporal ILF can be used with any renderer as long it is developed in a compatible way. Section 6.3.2 provides the schematic diagram of the various parts and the workflow of the Temporal ILF engine and how it may interact with a regular renderer.
- Temporal ILF has been implemented in a real-world scene with the methods described here and the relevant frames of a rendered video consists of a synthetic object lit by the light in the scene has been included.
- The limitations of the Temporal iLF has been identified, discussed and the future research directions has been suggested.

The next chapter will conclude the works that has been done in this thesis, their interconnections, applications, limitations and possible future directions.

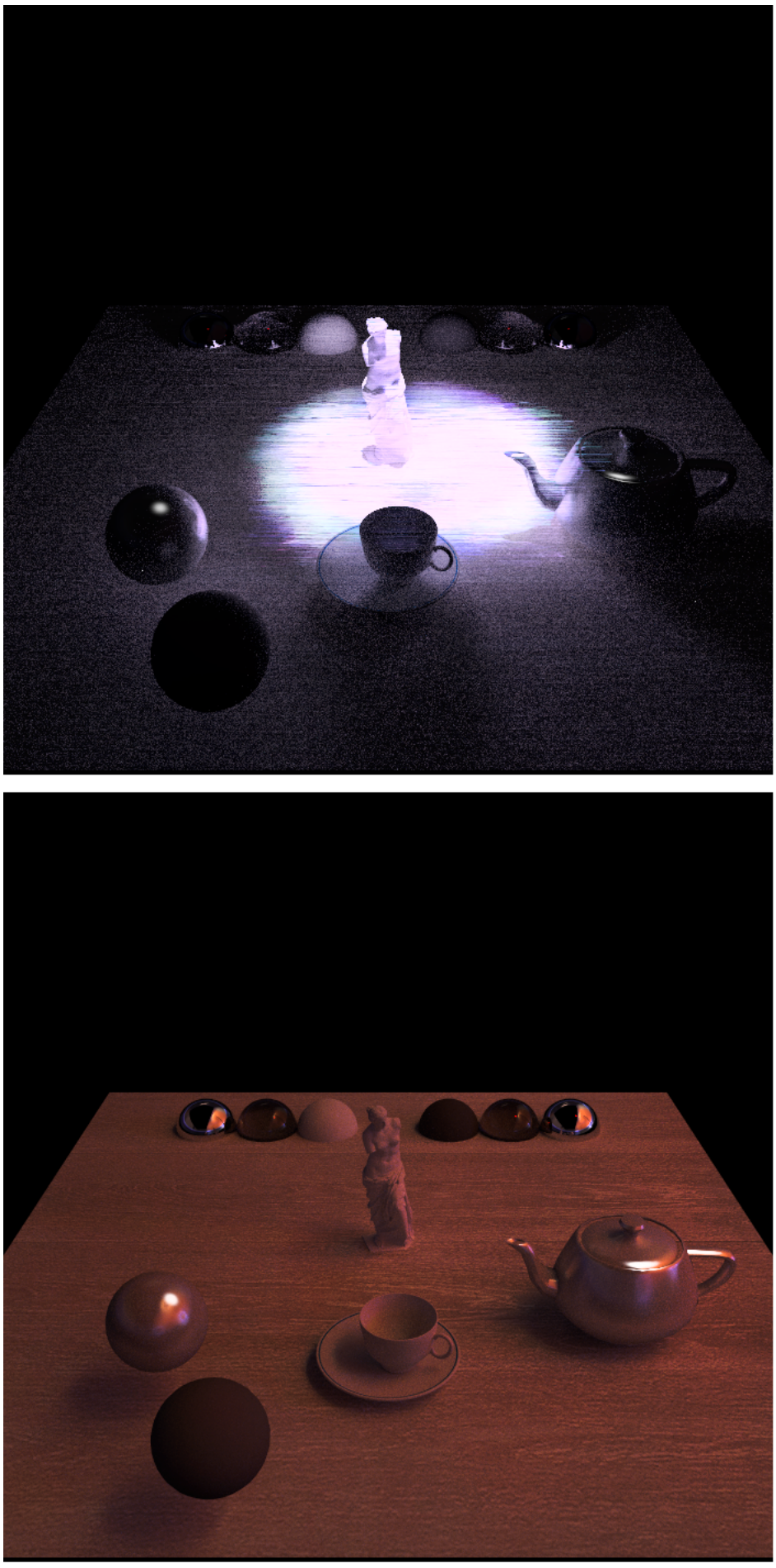


Figure 6.13: Only Spotlight and Color Bleeding from red wall

Chapter 7

Conclusion

"A poem is never finished; only abandoned."

- Paul Valerie

There are a number of applications that extensively use CGI to render synthetic objects in a real-world scene with complex light. Previously Section 1.1.2 and later Section 7.2 discusses a few such application areas such as the motion pictures, computer games industry, flight and car simulations, architecture, product advertisement, furniture industry, AR, VR, and many more. None of the present techniques are able to capture and render from spatio-temporally varying real-world light. This thesis presents a novel technique called Temporal ILF for rendering with spatially and temporally varying light. In the process, this research has done a thorough literature survey of the related techniques and has implemented a version of the ILF technique [Unger, 2009]. In addition, a novel technique called DCP has been developed which enables synthetic changes in the direct light sources in a ILF database with indirect light fidelity (see Chapter 5 for details).

This chapter concludes the work presented in this thesis by summarising the contributions, describing the possible applications, pointing out the limitations and suggesting a future directions of this research.

7.1 Summary of Contributions

To date, there are no techniques to capture and render synthetic scenes from complex real-world light which varies spatially and changes over time. This thesis took inspirations from the ILF technique which can only record spatial variance and developed the "Temporal ILF" for rendering with spatial and temporally variant lighting. The primary contributions of this thesis are:

- An implementation of a variant of the "free-form ILF" technique for rendering with high-fidelity spatially varying light. The implementation uses a flexible representation of the ray-database suitable for integration with other techniques such as the DCP. The ILF engine developed can be integrated with any renderer and is self-sustainable. (Chapter 4)

- A novel technique for dynamically changing the direct lights stored in ILF database synthetically with indirect light fidelity. The DCP technique developed in this thesis calculates the effect of any changes made in the light sources to the indirect lights reflected from the captured scene. (Chapter 5)

- A new method to render with dynamically changing, spatio-temporally variant real-world lighting. The Temporal ILF provides an end-to-end solution for rendering with dynamically changing complex lighting. (Chapter 6)

7.2 Applications

Providing means to render with dynamically changing complex real-world light has a number of applications such as motion pictures, computer games, product advertisements, architectural simulations for artistic and engineering and manufacturing applications, and vehicle simulations. The primary contribution of this thesis is the Temporal ILF technique has large potential application in the motion picture industry. The photorealistic renders of the CGI scenes in the feature films are very rarely rendered with real-world spatially varying light primarily due to the lack of a robust technique. Currently, apart from the ILF there are no other such techniques which has similar capabilities. As mentioned in Chapter 4, the ILF however, cannot render from dynamically changing complex lighting. The proposed Temporal ILF, on the other hand, is capable of capturing and rendering from complex real-world light which may dynamically change over time.

Apart from the motion pictures, the computer games industry has a huge demand for photorealistic renderings with complex lighting. Traditionally, artificial lighting has been typically used for their renderings. However, there is a surge in using real-world lights in computer games especially in the wake of the VR and AR enabled games. Temporal ILF provides a means of capturing and rendering with complex, dynamic lighting for the computer games applications.

There are several other applications that benefit from rendering with dynamic and complex lighting. In architecture, artistic renderings of ancient historical places are excellent tool for historians, anthropologists as well as common people. In modern architecture, photorealistic renderings of complex buildings are almost crucial , as well as several manufacturing and engineering applications.

Showcasing commercial products such as furnitures, accessories and related products often use complex lighting as well. Providing them with means to render with real world light will enable the production of highly customised and personalised renderings. There are already applications that exist, such as ARkit, which enables the user to capture and reproduce personal environments such as a living room. These applications can be used to preview how a piece of furniture might look like in a specific room, at a specific time of the day under specific lighting conditions.

Finally, the DCP technique has been primarily developed to aid the Temporal ILF technique. It however has its own applications outside Temporal ILF. One such application can be object relighting. Provided with an ILF capture of an environment and a 3D point-cloud, a still picture of an object can be relit by changing the light positions as DCP can calculate the contributions of incoming direct light from the ILF.

The existing techniques for these applications use simpler IBL based techniques for the photorealistic renders. The Temporal ILF, although a costlier option than the rest, will bring the photorealism into a new, higher standard. Currently a limitation of this technique is the costly, labour intensive capture technique. In the future, with a scalable implementation of Temporal ILF will see a much cost effective solution for many commercial applications.

7.3 Limitations

There are a few limitations of the works presented in this thesis. Although most of them have been presented in the respective chapters, this section summarises the primary limitations.

7.3.1 Capture:

As mentioned in Chapter4, capturing ILF has always been a complex and resource dependent task. The easiest way to capture ILF is using HDR video cameras. Unfortunately, such cameras are still prohibitively expensive and largely inaccessible to the general consumers. Moreover, these cameras are also not portable, making them harder to move around the capture area without a capture rig such as the one mentioned in Chapter 4. The still HDR light probe method, which has been used to capture ILF throughout this work, is an extremely slow and cumbersome method of capturing an ILF. A typical capture time in a (2 Meters $\times$ 2 Meters) table top can be as long as 6 hours with moderately dense spatial resolution.

The complexity and the resource requirement of the capture is one of the main reason that ILF is not being as widely used as some other techniques with much simpler capture methods, such as IBL. The Temporal ILF also suffers from

the similar limitations because it requires a base ILF capture as a part of its capturing pipeline.

7.3.2 Tracking Light Sources:

Temporal ILF requires the light sources to be tracked throughout the capture detecting the position, colour and the intensity at any given time. While tracking the position of the light sources has been implemented successfully, detecting the change in the light intensity automatically still depends on HDR video.

As there was no HDR video camera available during the development of this work, intensity changes in light sources were manually recorded with a light-meter as described in Chapter 6. This is a major limitation of the current work which needs to be addressed in order to facilitate potential applications to take advantage of the Temporal ILF technique.

7.3.3 Material Properties of Scene Boundaries:

The DCP technique estimates the change in the indirect lighting by calculating the direct light reflecting from the UPoints (described in Section 5.2.2) along the scene geometry. Although the UPoints can be assigned with any BRDF model, by default it assumes diffuse BRDF which may not be close to the real BRDF of the part of the scene in the real-world capture. This is a minor issue as there are methods that already exist to approximately capture the BRDF and obtain the closest model of a point in a surface with the respective part of the geometry manually.

Measuring and estimating BRDFs is a complex research area in itself and thus was out of the scope of the present work. The ability to measure BRDFs has not been included in the DCP technique, although it could be added as an additional module. This is an implementational limitation that can be added with improved, future implementations of DCP and Temporal ILF.

7.3.4 Scalability

Despite the the successful implementations of ILF by Unger et al and subsequently, the Temporal ILF presented here, a primary limitation is scalability. ILF is meant for small indoor scenes. Capturing large spaces especially outdoors will result in a significant memory overhead as the result of a very large ray-database. This has been discussed in Chapter 4 and Chapter 6. The present solutions do not offer any effective compression methods which can make the size of the ILF ray-database manageable. For large scale applications of dynamic, real-world light capture, such as the virtual reality vehicle testing for the automotive and aeronautics industry, the apparent lack of scalability in ILF and in Temporal ILF is a major hindrance. Future

implementations of Temporal ILF should explore ways to effectively compress ILF ray-databases in order to capture and render with light in any arbitrarily large space without a significant resource overhead.

7.4 Future Directions

Identifying the limitations of this work has opened several new opportunities for future research and development. Below are the possible future research ideas to make Temporal ILF more accessible to the end user and thereby attracting various industries to build commercial applications by using this technique.

7.4.1 Portable Capturing Technique

Chapter 4 and Chapter 6 has identified that the primary limitation of a consumer grade ILF based application is its capture complexity. Affordable HDR cameras and HDR sensors are still very scarce which makes it inaccessible to most users. A potential research direction addressing this will be to reverse-tonemap [Banterle et al., 2006; Rempel et al., 2007] LDR pictures intelligently producing high-fidelity HDR light probes. Machine learning approaches can be useful to determine different parts of the scene to intelligently control the exposure on-the-fly in order to capture complex lighting and still produce high-fidelity light probes.

Tracking the position of the light probes is another issue with freely moving capturing methods. This thesis implemented a simple and effective technique based on ArUco markers. Similar methods can be used to automatically detect the frame positions and other non-computer vision based tracking methods could be explored.

7.4.2 Scalable Temporal ILF

As mentioned before, scalability is a persistent issue with all ILF based techniques. Potential future research could be done in the direction of an end to end, scalable Temporal ILF. A starting point whould be further research on ray-database compression to reduce memory overhead during rendering. A robust capture technique will also be useful for capturing large spaces such as outdoors. There are affordable commercial drones which can be used in tandem with the portable capturing device discussed above to capture large scale ILF. Subsequently, as the usual tracking technique based on marker detection would be useless in outdoors, other means of position detections must also be explored.

Figure 7.1 presents a process flowchart for the ongoing future works . The first step is to develop the compression technique for Temporal ILFs and adding more complex BRDFs to it. The next step will be to develop the portable capture technique after

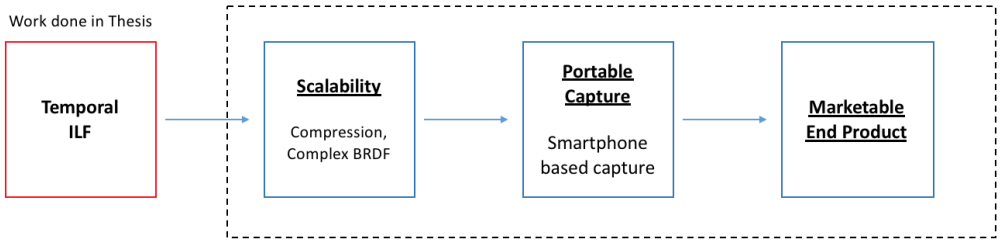


Figure 7.1: A process flow for the future works.

which Temporal ILF could be made into an end product for various commercial applications.

7.5 Final Words

This work proposed a novel technique to capture and render from complex and dynamically changing real-world light which paves a new way not trodden before. The evaluations using the controlled virtual environments showed the high accuracy of the proposed techniques. The real-world implementation produced photorealistic renderings with real-world light. Although the works presented here is self-sufficient, various resource constraints, however, prevented this Temporal ILF being implemented in large scale commercial-grade applications at the present time. Luckily, there are already some progresses into the future ideas mentioned in this chapter.

This thesis ends with the hope that the Temporal ILF technique has provided a firm foundation which will further evolve into a practical, end-to-end solution for rendering with complex, real-world light in the near future.

Bibliography

FARO 3D Laser Scanner . http://www.faro.com/products/3d-surveying/laser-scanner-faro-focus-3d/overview. [Overview of Faro 3D Scanner. Online; accessed 7-April-2015].

Computer Games Revenue in 2018. https://www.statista.com/statistics/292056/video-game-market-value-worldwide/. [Statistical data on the revenue in computer games. Online; accessed 12-June-2018].

Edward H Adelson and James R Bergen. The plenoptic function and the elements of early vision. *Computational models of visual processing*, 1(2):3–20, 1991.

Frederik Anrys and Philip Dutré. Image-based lighting design. 2004.

AB Arons and MB Peppard. Einstein's proposal of the photon concept—a translation of the annalen der physik paper of 1905. *American Journal of Physics*, 33(5):367–374, 1965.

Francesco Banterle, Patrick Ledda, Kurt Debattista, and Alan Chalmers. Inverse tone mapping. In *Proceedings of the 4th international conference on Computer graphics and interactive techniques in Australasia and Southeast Asia*, pages 349–356. ACM, 2006.

Francesco Banterle, Alessandro Artusi, Kurt Debattista, and Alan Chalmers. *Advanced high dynamic range imaging*. CRC press, 2017.

FO Bartell, EL Dereniak, and WL Wolfe. The theory and measurement of bidirectional reflectance distribution function (brdf) and bidirectional transmittance distribution function (btdf). In *Radiation scattering in optical systems*, volume 257, pages 154–161. International Society for Optics and Photonics, 1981.

James F. Blinn. Models of light reflection for computer synthesized pictures. *SIGGRAPH Comput. Graph.*, 11(2):192–198, July 1977. ISSN 0097-8930. doi: 10.1145/965141.563893. URL http://doi.acm.org/10.1145/965141.563893.

James F Blinn and Martin E Newell. Texture and reflection in computer generated images. *Communications of the ACM*, 19(10):542–547, 1976.

Eric Braaten and George Weller. An improved low-discrepancy sequence for multidimensional quasi-monte carlo integration. *Journal of Computational Physics*, 33 (2):249–258, 1979.

Emilio Camahort, Francisco Abad, and Don Fussell. A line-space analysis of light-field representations. *Graphical Models*, 71(5):169–183, 2009.

Shenchang Eric Chen. Quicktime vr: An image-based approach to virtual environment navigation. In *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, pages 29–38. ACM, 1995.

Yong-Sheng Chen, Yi-Ping Hung, and Chiou-Shann Fuh. Fast block matching algorithm based on the winner-update strategy. *IEEE Transactions on Image Processing*, 10(8):1212–1222, 2001.

Michael F Cohen and Donald P Greenberg. The hemi-cube: A radiosity solution for complex environments. In *ACM SIGGRAPH Computer Graphics*, volume 19, pages 31–40. ACM, 1985.

Michael F Cohen and John R Wallace. *Radiosity and realistic image synthesis*. Elsevier, 2012.

R. L. Cook and K. E. Torrance. A reflectance model for computer graphics. *ACM Trans. Graph.*, 1(1):7–24, January 1982. ISSN 0730-0301. doi: 10.1145/357290.357293. URL `http://doi.acm.org/10.1145/357290.357293`.

Kurt Debattista, Thomas Bashford-Rogers, Carlo Harvey, Brian Waterfield, and Alan Chalmers. Subjective evaluation of high-fidelity virtual environments for driving simulations. *IEEE Transactions on Human-Machine Systems*, 2017.

Paul Debevec. Image-based lighting. *IEEE Computer Graphics and Applications*, 22 (2):26–34, 2002.

Paul Debevec. Rendering synthetic objects into real scenes: Bridging traditional and image-based graphics with global illumination and high dynamic range photography. In *ACM SIGGRAPH 2008 classes*, page 32. ACM, 2008.

Paul Debevec, Tim Hawkins, Chris Tchou, Haarm-Pieter Duiker, Westley Sarokin, and Mark Sagar. Acquiring the reflectance field of a human face. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 145–156. ACM Press/Addison-Wesley Publishing Co., 2000.

Philip Dutre, Kavita Bala, Philippe Bekaert, and Peter Shirley. *Advanced Global Illumination*. AK Peters Ltd, 2006. ISBN 1568813074.

A Einstein. Über einen die erzeugung und verwandlung des lichtes betreffenden heuristischen gesichtspunkt [adp 17, 132 (1905)]. *Annalen der Physik*, 14(S1): 164–181, 2005.

James A Ferwerda, Sumanta N Pattanaik, Peter Shirley, and Donald P Greenberg. A model of visual adaptation for realistic image synthesis. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 249–258. ACM, 1996.

Richard P Feynman, Robert B Leighton, and Matthew Sands. *The Feynman lectures on physics, Vol. I: The new millennium edition: mainly mechanics, radiation, and heat*, volume 1. Basic books, 2011.

Richard Phillips Feynman and A Zee. *QED: The strange theory of light and matter*. Princeton University Press, 2006.

Alain Fournier, Atjeng S Gunawan, and Chris Romanzin. Common illumination between real and computer generated scenes. In *Graphics Interface*, pages 254–254. CANADIAN INFORMATION PROCESSING SOCIETY, 1993.

Ivar Fredholm. Sur une classe d'équations fonctionnelles. *Acta Mathematica*, 27(1): 365–390, 1903. ISSN 1871-2509. doi: 10.1007/BF02421317. URL `http://dx.doi.org/10.1007/BF02421317`.

Martin Fuchs, Volker Blanz, and Hans-Peter Seidel. Bayesian relighting. In *Rendering Techniques*, pages 157–164, 2005.

Sergio Garrido-Jurado, Rafael Muñoz-Salinas, Francisco José Madrid-Cuevas, and Manuel Jesús Marín-Jiménez. Automatic generation and detection of highly reliable fiducial markers under occlusion. *Pattern Recognition*, 47(6):2280–2292, 2014.

Abhijeet Ghosh, Arnaud Doucet, and Wolfgang Heidrich. Sequential sampling for dynamic environment map illumination. In *Rendering Techniques*, pages 115–126, 2006.

Steven J Gortler, Radek Grzeszczuk, Richard Szeliski, and Michael F Cohen. The lumigraph. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 43–54. ACM, 1996.

William Rowan Hamilton. Theory of systems of rays. *Transactions of the Royal Irish Academy*, 15(1828):69–174, 1828.

Pat Hanrahan and Wolfgang Krueger. Reflection from layered surfaces due to subsurface scattering. In *Proceedings of the 20th annual conference on Computer graphics and interactive techniques*, pages 165–174. ACM, 1993.

Chris Harris and Mike Stephens. A combined corner and edge detector. In *Alvey vision conference*, volume 15, pages 10–5244. Citeseer, 1988.

Vlastimil Havran, Miloslaw Smyk, Grzegorz Krawczyk, Karol Myszkowski, and Hans-Peter Seidel. Interactive system for dynamic scene lighting using captured video environment maps. In *Rendering Techniques*, pages 31–42, 2005.

Ivo Ihrke, Timo Stich, Heiko Gottschlich, Marcus Magnor, and Hans-Peter Seidel. Fast incident light field acquisition and rendering. 2008.

Aaron Isaksen, Leonard McMillan, and Steven J Gortler. Dynamically reparameterized light fields. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 297–306. ACM Press/Addison-Wesley Publishing Co., 2000.

Shahram Izadi, David Kim, Otmar Hilliges, David Molyneaux, Richard Newcombe, Pushmeet Kohli, Jamie Shotton, Steve Hodges, Dustin Freeman, Andrew Davison, et al. Kinectfusion: real-time 3d reconstruction and interaction using a moving depth camera. In *Proceedings of the 24th annual ACM symposium on User interface software and technology*, pages 559–568. ACM, 2011.

James T Kajiya. The rendering equation. In *ACM Siggraph Computer Graphics*, volume 20, pages 143–150. ACM, 1986.

Alexander Keller. Instant radiosity. In *Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, pages 49–56. ACM Press/Addison-Wesley Publishing Co., 1997.

Ian R Kenyon. *The light fantastic: a modern introduction to classical and quantum optics*. Oxford University Press, USA, 2008.

Thomas Kollig and Alexander Keller. Efficient multidimensional sampling. In *Computer Graphics Forum*, volume 21, pages 557–563. Wiley Online Library, 2002.

Jaroslav Krivanek, Pascal Gautron, Sumanta Pattanaik, and Kadi Bouatouch. Radiance caching for efficient global illumination computation. *IEEE Transactions on Visualization and Computer Graphics*, 11(5):550–561, 2005.

Gaetano C La Delfa, Vincenzo Catania, Salvatore Monteleone, Juan F De Paz, and Javier Bajo. Computer vision based indoor navigation: A visual markers evaluation. In *Ambient Intelligence-Software and Applications*, pages 165–173. Springer, 2015.

Eric P Lafortune and Yves D Willems. Bi-directional path tracing. 1993.

Johann Heinrich Lambert. *Photometria sive de mensura et gradibus luminis, colorum et umbrae*. Klett, 1760.

Gregory Ward Larson, Holly Rushmeier, and Christine Piatko. A visibility matching tone reproduction operator for high dynamic range scenes. *IEEE Transactions on Visualization and Computer Graphics*, 3(4):291–306, 1997.

Marc Levoy. Light fields and computational imaging. *Computer*, 39(8):46–55, 2006.

Marc Levoy and Pat Hanrahan. Light field rendering. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 31–42. ACM, 1996.

Marc Levoy, Kari Pulli, Brian Curless, Szymon Rusinkiewicz, David Koller, Lucas Pereira, Matt Ginzton, Sean Anderson, James Davis, Jeremy Ginsberg, et al. The digital michelangelo project: 3d scanning of large statues. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 131–144. ACM Press/Addison-Wesley Publishing Co., 2000.

Siwei Lyu and Hany Farid. How realistic is photorealistic? *IEEE Transactions on Signal Processing*, 53(2):845–850, 2005.

Stephen R Marschner, Stephen H Westin, Eric PF Lafortune, Kenneth E Torrance, and Donald P Greenberg. Image-based brdf measurement including human skin. In *Rendering Techniques' 99*, pages 131–144. Springer, 1999.

Vincent Masselus, Pieter Peers, Philip Dutré, and Yves D Willems. Relighting with 4d incident light fields. In *ACM Transactions on Graphics (TOG)*, volume 22, pages 613–620. ACM, 2003.

J. Clerk Maxwell. A dynamical theory of the electromagnetic field. *Philosophical Transactions of the Royal Society of London*, 155:459–512, 1865. ISSN 02610523. URL `http://www.jstor.org/stable/108892`.

Joshua McNamee, Jonathan Hatchett, Kurt Debattista, and Alan Chalmers. Live hdr video streaming on commodity hardware. *SPIE Proceedings: Applications of Digital Image Processing XXXVIII*, 9599:95990U, 2015.

Nicholas Metropolis and Stanislaw Ulam. The monte carlo method. *Journal of the American statistical association*, 44(247):335–341, 1949.

Benjamin Mora, Jean Pierre Jessel, and René Caubet. A new object-order ray-casting algorithm. In *Proceedings of the conference on Visualization'02*, pages 203–210. IEEE Computer Society, 2002.

Ratnajit Mukherjee, Kurt Debattista, Thomas Bashford-Rogers, Brian Waterfield, and Alan Chalmers. A study on user preference of high dynamic range over low dynamic range video. *The Visual Computer*, 32(6-8):825–834, 2016.

Eihachiro Nakamae, Koichi Harada, Takao Ishizaki, and Tomoyuki Nishita. A montage method: The overlaying of the computer generated images onto a background photograph. In *ACM SIGGRAPH Computer Graphics*, volume 20, pages 207–214. ACM, 1986.

Ren Ng, Marc Levoy, Mathieu Brédif, Gene Duval, Mark Horowitz, and Pat Hanrahan. Light field photography with a hand-held plenoptic camera. *Computer Science Technical Report CSTR*, 2(11), 2005.

Matt Pharr, Wenzel Jakob, and Greg Humphreys. *Physically based rendering: From theory to implementation*. Morgan Kaufmann, 2016.

Bui Tuong Phong. Illumination for computer generated pictures. *Commun. ACM*, 18(6):311–317, June 1975. ISSN 0001-0782. doi: 10.1145/360825.360839. URL `http://doi.acm.org/10.1145/360825.360839`.

Max Planck. Ueber die elementarquanta der materie und der elektricität. *Annalen der Physik*, 309(3):564–566, 1901.

Laiyun Qing, Shiguang Shan, Wen Gao, and Bo Du. Face recognition under generic illumination based on harmonic relighting. *International Journal of Pattern Recognition and Artificial Intelligence*, 19(04):513–531, 2005.

Allan G Rempel, Matthew Trentacoste, Helge Seetzen, H David Young, Wolfgang Heidrich, Lorne Whitehead, and Greg Ward. Ldr2hdr: on-the-fly reverse tone mapping of legacy video and photographs. In *ACM transactions on graphics (TOG)*, volume 26, page 39. ACM, 2007.

Szymon Rusinkiewicz, Olaf Hall-Holt, and Marc Levoy. Real-time 3d model acquisition. *ACM Transactions on Graphics (TOG)*, 21(3):438–446, 2002.

Imari Sato, Yoichi Sato, and Katsushi Ikeuchi. Acquiring a radiance distribution to superimpose virtual objects onto a real scene. *IEEE transactions on visualization and computer graphics*, 5(1):1–12, 1999.

Atsushi Takagi. Ray tracing method, September 2 1997. US Patent 5,663,789.

Jonas Unger. Incident light fields. 2009.

Jonas Unger, Andreas Wenger, Tim Hawkins, Andrew Gardner, and Paul Debevec. Capturing and rendering with incident light fields. Technical report, DTIC Document, 2003.

Jonas Unger, Stefan Gustavson, Mark Ollila, and Mattias Johannesson. A real time light probe. In *The 25th Eurographics Annual Conference 2004 Short papers and Interactive Applications, Grenoble, France*, 2004.

Jonas Unger, Stefan Gustavson, and Anders Ynnerman. Spatially varying image based lighting by light probe sequences. *The Visual Computer*, 23(7):453–465, 2007.

Jonas Unger, Stefan Gustavson, Per Larsson, and Anders Ynnerman. Free form incident light fields. In *Computer Graphics Forum*, volume 27, pages 1293–1301. Wiley Online Library, 2008.

Eric Veach and Leonidas J Guibas. Metropolis light transport. In *Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, pages 65–76. ACM Press/Addison-Wesley Publishing Co., 1997.

Hermann Von Helmholtz. *Handbuch der physiologischen Optik*, volume 9. Voss, 1867.

Yang Wang, Lei Zhang, Zicheng Liu, Gang Hua, Zhen Wen, Zhengyou Zhang, and Dimitris Samaras. Face relighting from a single image under arbitrary unknown lighting conditions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(11):1968–1984, 2009.

Gregory J. Ward. Measuring and modeling anisotropic reflection. *SIGGRAPH Comput. Graph.*, 26(2):265–272, July 1992. ISSN 0097-8930. doi: 10.1145/142920.134078. URL `http://doi.acm.org/10.1145/142920.134078`.

Gregory J Ward, Francis M Rubinstein, and Robert D Clear. A ray tracing solution for diffuse interreflection. *ACM SIGGRAPH Computer Graphics*, 22(4):85–92, 1988.

Turner Whitted. An improved illumination model for shaded display. *Communications*, 1980.

Turner Whitted. An improved illumination model for shaded display. In *ACM Siggraph 2005 Courses*, page 4. ACM, 2005.

Tien-Tsin Wong, Wai-Shing Luk, and Pheng-Ann Heng. Sampling with hammersley and halton points. *Journal of graphics tools*, 2(2):9–24, 1997.

Jacob Ziv and Abraham Lempel. Compression of individual sequences via variable-rate coding. *IEEE transactions on Information Theory*, 24(5):530–536, 1978.